20

COMMUNICATIONS

# Local area networks

## What is an LAN?

The digital PABX (as we saw in *Communications 19*) provides a mechanism for data transmission between a number of digital devices over an existing, internal communications network. This PABX data network is only one of many types of **local area network**, or LAN, (*figure 1*).

There are three basic categories of LAN: non-switched; circuit-switched; and packet switched. A PABX forms the switching control unit of a **circuit switched** data network. **Non-switched** LANs comprise a number of data devices, permanently connected in one of two ways: either each device is connected to every other device; or devices are connected in a hierarchy.

By far the most important category of LAN, however, is formed by the basic **packet switching** communications methods we have seen in earlier chapters. Within this category, many different formats of connection are possible – the four most common being shown in *figure 1*. These different network **topologies** (or shapes) all share some basic similarities:

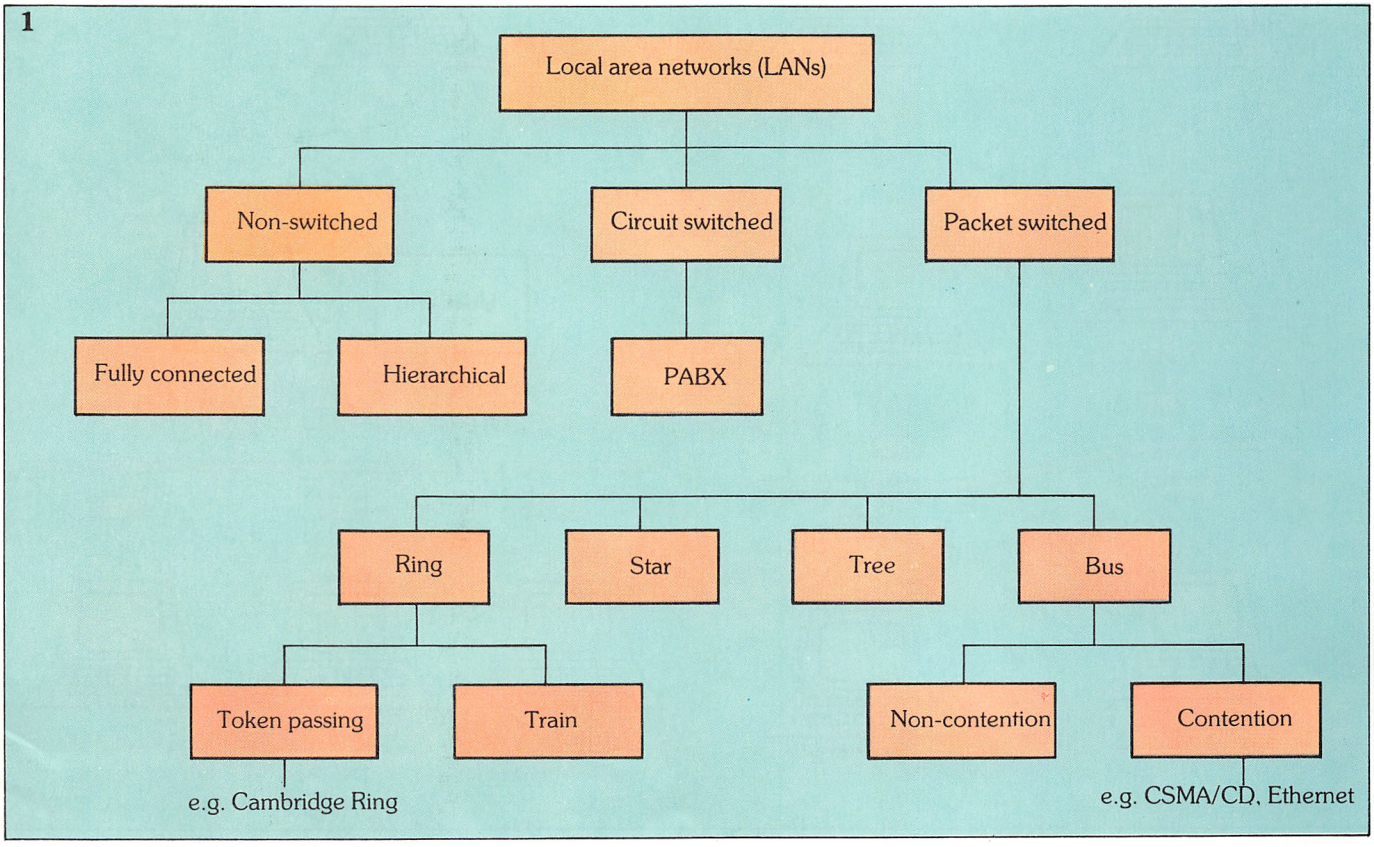
**1)** Data is transmitted at very high data signalling rates, i.e. between about 0.1 Mbits $s^{-1}$ and 10 Mbits $s^{-1}$.

**2)** The maximum geographical area for an LAN is about 1 $km^2$.

**3)** The product of any particular LAN's cable length and data signalling rate is more or less constant. For example, for a particular LAN of cable length 1 km and data signalling rate 1 Mbits $s^{-1}$, the length/bit-rate product will be 1 km Mbits $s^{-1}$. If

**1. There are three** distinct types of local area network: non-switched; circuit switched; and packet switched.

1



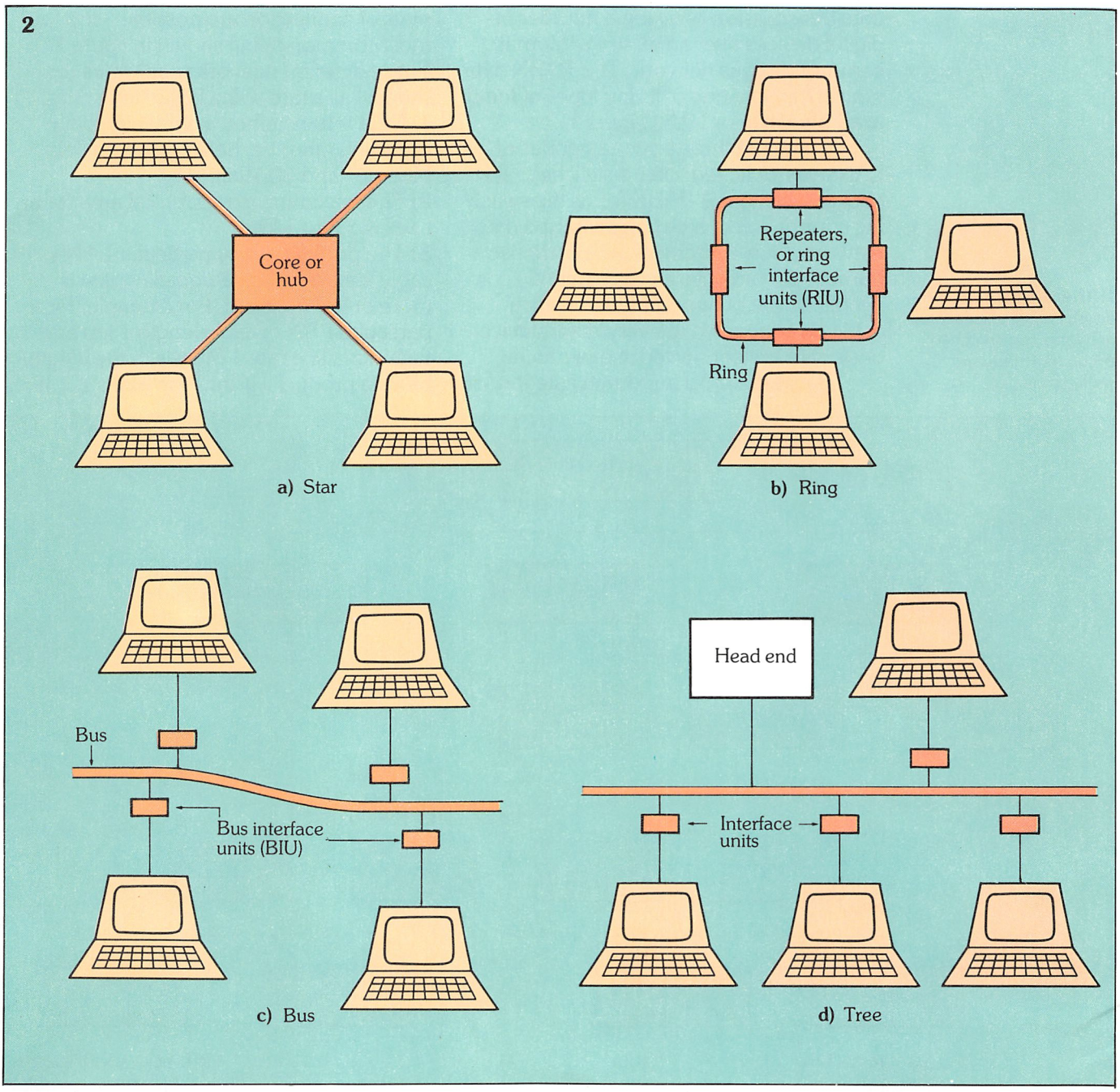
the cable length for a certain application were to be increased to, say, 2 km, then for a length/bit-rate product of 1 km Mbits $s^{-1}$, the bit-rate is limited to 0.5 Mbits $s^{-1}$. Alternatively, if the cable length is limited to 0.5 km, then the bit-rate is increased to 2 Mbits $s^{-1}$.

**4)** LANs usually use a dedicated transmission medium, e.g. coaxial cable or optical fibre.

All packet switched LANs also transmit data in basically the same way. Data traffic generated by DTE such as microprocessor based computers, terminals, host computers etc., is usually in short bursts separated by comparatively long idle periods. During bursts, the data signalling rate may be as low as 300 bits $s^{-1}$ or as high as several Mbits $s^{-1}$; during idle periods the data signalling rate is, of course, zero.

This contrasts with the fixed data signalling rate of a digital PABX based data network, i.e. 64,000 bits $s^{-1}$.

**2. Common physical topologies for LANs.**



## LAN formats

The formats of packet switched LANs vary in three main aspects: **topology**; **access method** (i.e. protocols); and the **transmission medium** used.

### Physical topologies

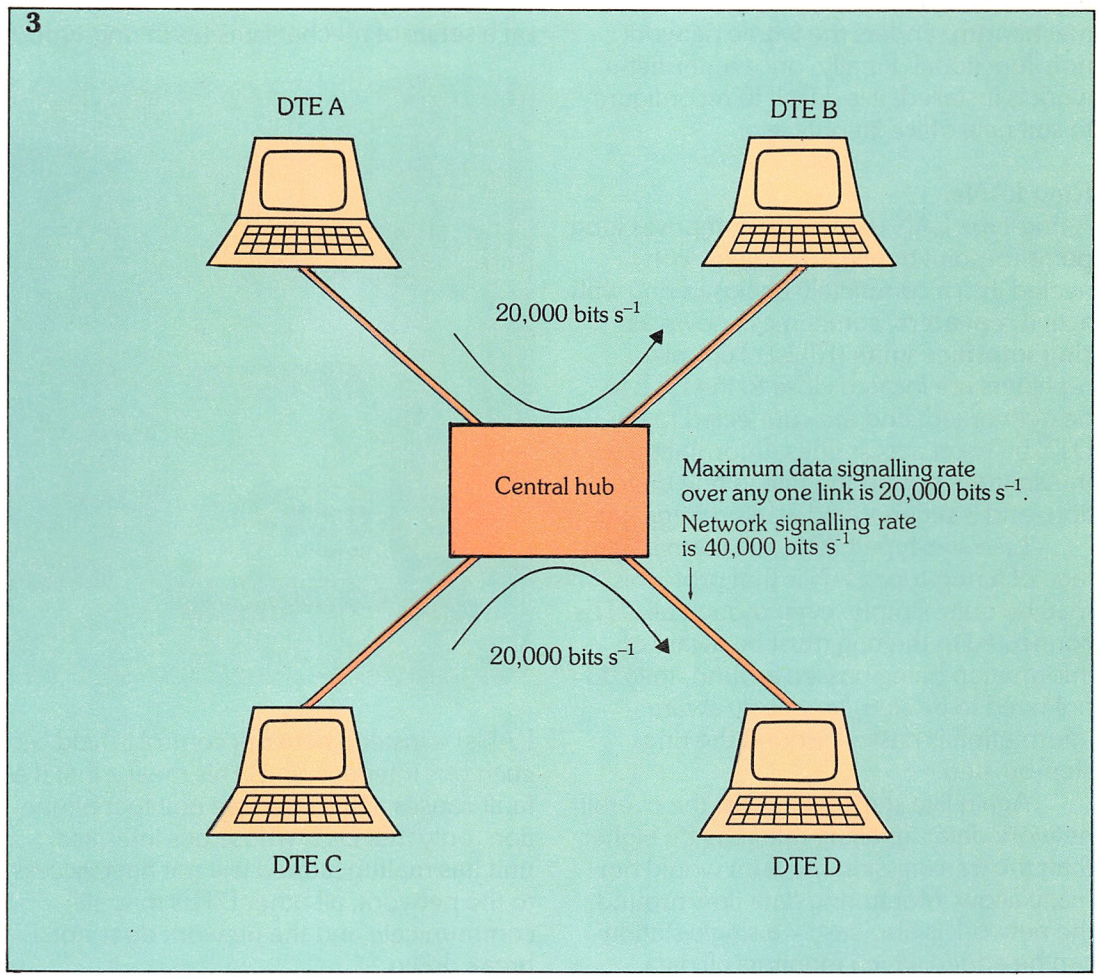
The four most common physical topologies used in LAN design – **star**, **ring**, **bus** and **tree** – are shown in *figure 2*.

A star-type LAN (*figure 2a*) comprises point-to-point transmission links carrying data to and from a **central hub** or core which contains some form of controlling mechanism, common to all connected devices. This is a similar arrangement to the star network of a PABX based internal telephone network.

Such a simple network has a number of significant advantages. First, the protocols required may be quite simple: data being transmitted from one DTE to another is simply routed by the central controlling mechanism from point-to-point – only those two DTE involved in the data transfer will need to be aware of what's going on.

Second, because information transfers do not involve all parts of the network, the overall network data rate may be *higher* than the bandwidth of the transmission medium would normally allow. To illustrate this, we'll take the example of the star network connected as shown in *figure 3* – DTE A transmits to DTE B, and DTE C transmits to DTE D. The bandwidth of the transmission medium allows, say, a maximum data signalling rate of 20,000 bits $s^{-1}$. So the data signalling rate between DTEs A and B, and C and D, is limited to this, over each link. But, the overall network data signalling rate is 40,000 bits $s^{-1}$ when both transmissions occur simultaneously.

The third and fourth advantages of a star-type LAN are derived from the use of




**3. The star-type LAN** has the advantage that the *overall* network data signalling rate may be higher than the bandwidth of the transmission medium would normally allow.

the central controlling mechanism: data flow in the network can be monitored every easily; and additional DTEs may be added, after installation, cheaply and easily.

Paradoxically, the central controlling mechanism also presents a number of system disadvantages. For example, the initial cost of installing the LAN is high: a complex central hub must always be purchased, regardless of the number of DTEs to be connected. Another disadvantage is that one small fault in the central mechanism renders the whole network non-functional. Finally, once a star network is installed, it is difficlt to reconfigure to suit new office layouts, etc.

### Ring LANs

A ring-type LAN (*figure 2b*) comprises long point-to-point transmission links, connected into a completely enclosed ring with active **repeaters**, sometimes known as **ring interface units** (RIUs). Typically, repeaters are located close to the DTE to be networked, and are connected to the DTE by short access lines. Information is transmitted around the ring in one direction, and is regenerated at every repeater.

Like star-type LANs, the main advantage of a ring-type LAN is that protocols may be quite simple: even though all DTEs connected to the ring must be aware of information being passed around, they do not need to be *simultaneously* aware – information is passed around the ring step-by-step.

Again like star-type LANs, the overall network data signalling rate may be higher than the transmission medium would normally allow. Monitoring data flow around the network is also easy – a single station can be added which monitors all data passing that point.

Another advantage is that the installation of a small ring network can be fairly inexpensive as no large controlling mechanism is required. However, if additional DTEs are to be added to the network after installation, complex (and hence costly) active repeaters must be added as well.

Like star LANs, a ring LAN suffers from the disadvantage that a single small fault in only one repeater may render the network inoperable.

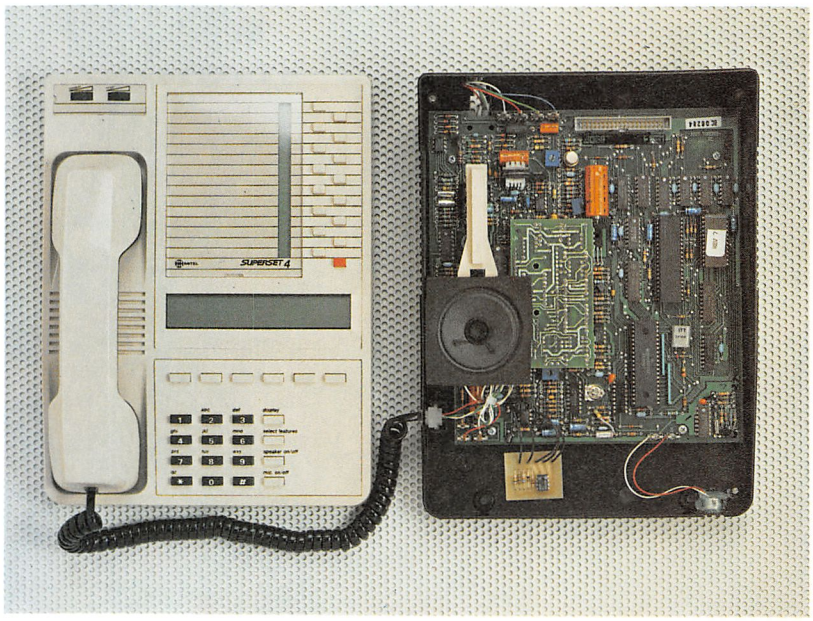
### Bus LANs

A bus-type LAN (*figure 2c*) consists of a single, long transmission medium. Access to the bus is accomplished via **active interfaces**, sometimes known as **bus interface units** (BIUs). Each bus interface unit is positioned close to a DTE, and connected to it with a short access line.

One of the biggest advantages of bus LANs is that they have what is known as **distributed control**. Control of the network is not held by a central mechanism (as in star-type LANs), nor is it dependent on a series of mechanisms (as in ring-type LANs) – instead network control is held by *each* bus interface unit. This means that if a fault causes a bus interface unit to malfunction, only that DTE whose bus interface unit has malfunctioned will not have access to the network; all other DTEs may still communicate and the network does not break down.

Another advantage of bus networks is that they may be easily reconfigured. Extending the transmission medium to wherever it is required to go, and connecting a bus interface unit, is all that is necessary to allow access by a DTE. Like ring-type LANs, however, the active interfaces to the network are complex and costly, so although reconfiguring the network to add extra DTEs is simple, it is an expensive process.

A final disadvantage of bus-type




**Below:** this Superset 4 messaging telephone is manufactured by MITEL and sold by British Telecom as the ST-14. The speaker (seen on the left) provides the loud speech facility. This telephone is based on the Motorola MC146805E2P microprocessor. (Photo: MITEL).

LANs is that complex network protocols are required.

### Tree LANs

A tree-type LAN (*figure 2d*) comprises a number of interconnected transmission links, arranged so that all information flow is directed towards a **head end** controlling mechanism. The head end then retransmits the information through the network to the required DTE.

From this description we can see that a tree-type LAN operates in a similar manner to a star-type LAN, having a single controlling mechanism through which all information flow is routed. However, its physical topology is not like a star network (i.e. point-to-point), but more like a bus network.

Tree-type LANs are therefore a kind of 'half-way house' between star and bus LANs, displaying the advantages and disadvantages of both. For example, reconfiguration of the network to include extra DTEs is a simple and cheap operation: the transmission medium is straightforward to extend (like that of bus LANs) and the extra interface units are also simple and cheap (like those of star LANs).

Initial installation costs are, on the other hand, quite high as a complex controlling mechanism – the head end – is required (like the central core of a star-type LAN).

### Logical topologies

The physical topologies of star, ring, bus and tree LANs can be viewed logically (i.e. with respect to the data flowing in the transmission medium) as belonging to one of only two topologies: the **broadcast bus**; or a **ring** with active repeaters.

The broadcast bus logical topology provides a transmission medium in which every piece of information present is detected by every DTE interfaced to the bus. Only those DTE specifically addressed by the information respond, however.

The ring network logical topology, like its corresponding physical topology, is formed by a sequence of separate transmission links arranged in a closed ring. Each piece of information is regenerated at each active repeater until it arrives at the DTE for which it is intended.

## Access methods

An access method is a protocol which ensures that:
1) only one DTE transmits at any one instant of time, or;
2) if more than one DTE is transmitting, then data is not corrupted or lost.

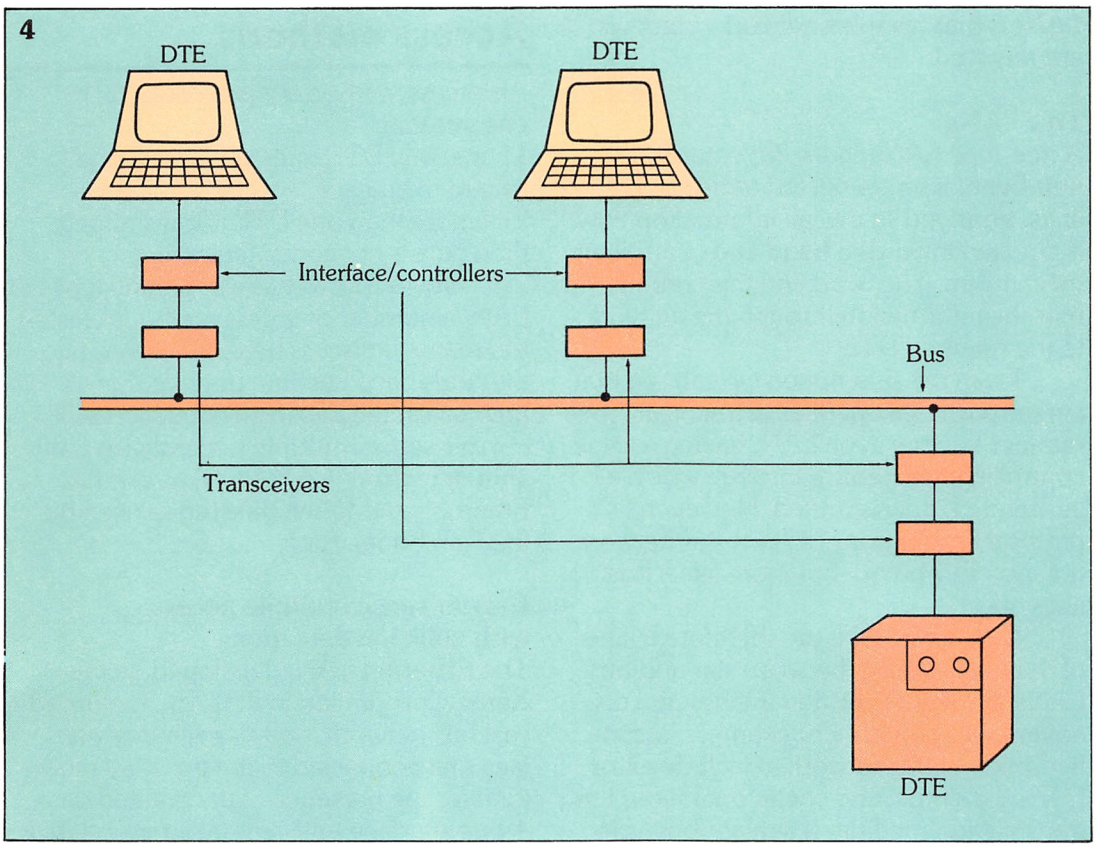
Obviously, as there are many types of LAN, a number of access methods exist – too many to discuss here. However, two methods have become popular due to their use in the two most common LANs: **carrier sense multiple access with collision detection** (CSMA/CD) as used in Ethernet; and **token passing** as used by the Cambridge Ring.

### Carrier sense multiple access with collision detection

The **Ethernet** LAN, developed by the Xerox Corporation in the U.S., is a broadcast bus network, and so each bus interface unit connected to the bus may monitor it for the presence of transmitted data. A bus interface unit can only transmit data if the bus is idle, i.e. if no data is presently being transmitted. If the bus is busy, data transmission onto the bus is deferred. As each bus interface unit has to contend with all others to transmit data, the CSMA/CD technique is known as a contention access method.

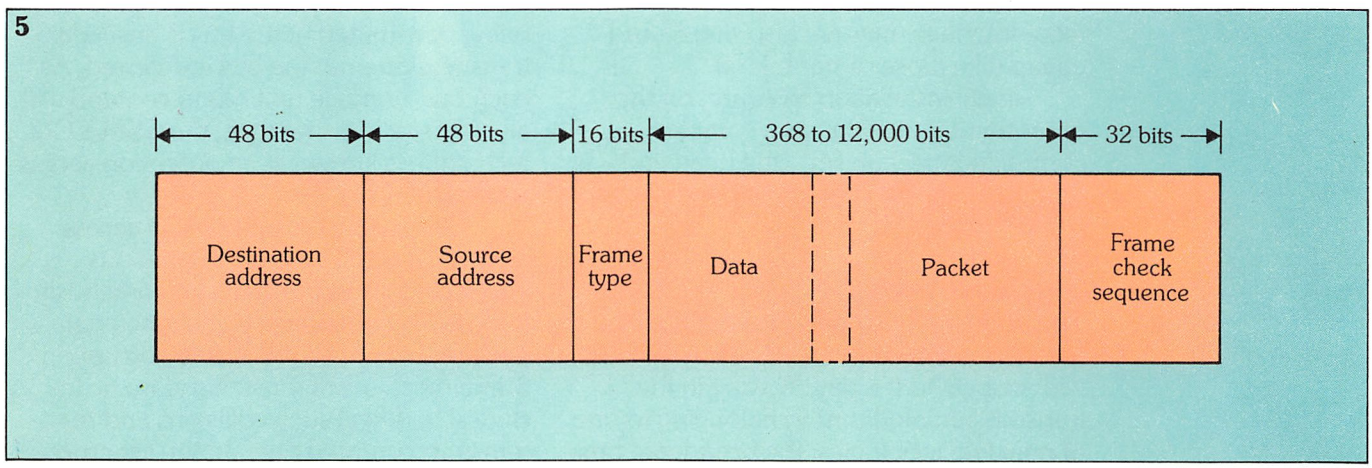
Even using the CSMA/CD access method, it is still possible for two bus interface units to transmit simultaneously – this causes **collision** of transmitted data packets and consequent data corruption. Some mechanism must therefore be included to detect such collisions and retransmit corrupted data. In the Ethernet CSMA/CD access method, data collisions are detected by each bus interface unit, causing them all to cease transmission for a short, random time. As the times during which all bus interface units do not transmit are random, when they begin transmissions again, data packets are automatically retransmitted at different times, so that collisions do not re-occur.

*Figure 4* shows a possible Ethernet network bus to which a number of DTEs are connected, with **transceivers** (i.e. transmitter/receivers) and **interface/controllers**. Transceivers handle all the




**4. The transceivers on the Ethernet** broadcast bus network handle all CSMA/CD protocol procedures ensuring that only complete packets are passed via the interface controller to the DTE.

**5. Ethernet frame format.**



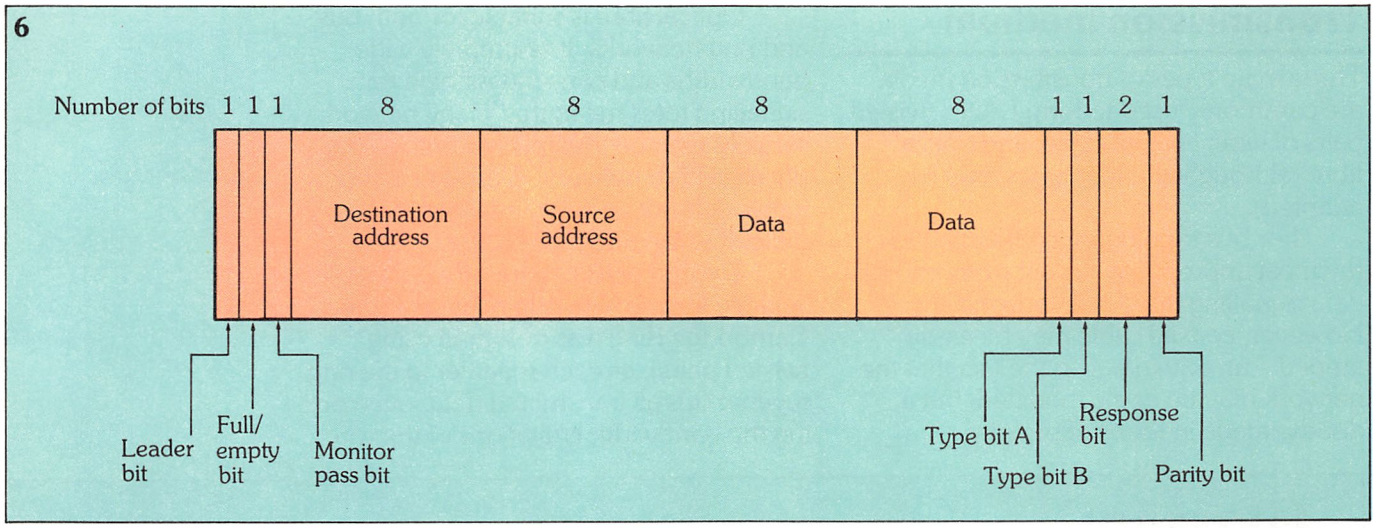

CSMA/CD protocol procedures so that only complete packets are passed, via the interface/controller, to the DTE.

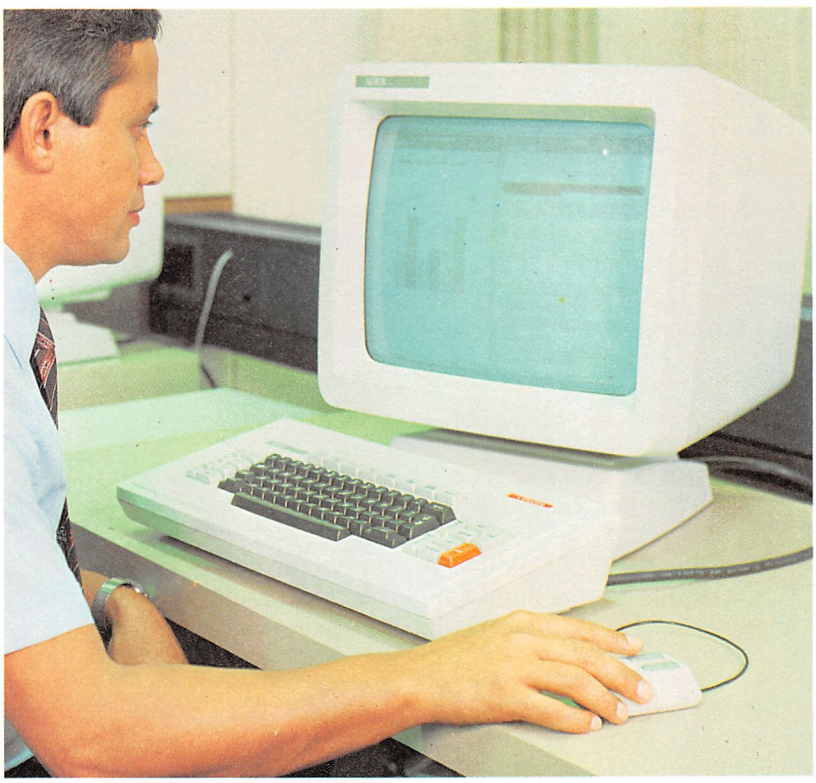
Data packets are incorporated by the transceiver into frames and a frame format for Ethernet is shown in *figure 5*. The use of these frames, together with CSMA/CD techniques, means that the data-link layer of the open systems interconnection model is more or less fulfilled. However, Ethernet does not *guarantee* error free delivery of data and so is not *completely* compatible with the ISO model.

Similarly, Ethernet cannot guarantee the successful transmission of data within a set time limit – it can only give a probability of this occurring, due to the use of CSMA/CD. Real time applications of Ethernet are therefore not suitable.

### Token passing

The **Cambridge Ring**, as its name suggests, is a logical ring-type network. Data is transmitted around the ring in **mini-packets** of about 40 bits. Essentially, mini-packets are continuously passed



**6. Cambridge Ring** mini-packet format.

**Above:** this Xerox 8010 executive workstation is linked by Ethernet to the Xerox 8000 network system.
(Photo: Rank Xerox (U.K.) Ltd/Namemakers Limited).

around the ring whether data is present in them or not. Any DTE which wishes to transmit data waits for an empty mini-packet to be received, inserts its data (together with control information), marks the mini-packet as full, and retransmits it.

The mini-packet is regenerated at every repeater in the ring, until the repeater of the called DTE receives it. This repeater copies the data to the DTE, marks the mini-packet as being accepted, then retransmits it.

When the repeater connected to the calling DTE receives this acknowledged mini-packet, it marks the mini-packet as empty, then retransmits it for use by other DTEs. A Cambridge Ring mini-packet format is shown in *figure 6*.

An 8-bit field is used to indicate the destination address, so up to 256 (i.e. $2^8$) DTEs may be connected to a ring. Similarly, an 8-bit field is used to indicate the source address. Two 8-bit fields contain data.

The remainder of the bits in the mini-packet are used to hold control and status information. The important bits are a **full/empty bit**, which indicates whether data is held in the mini-packet or not, and two bits which indicate the called DTE's acknowledgement response.

It is the process of constantly passing mini-packets around the ring which earns the access method its name of **token passing**. Another name sometimes used, even more descriptive of the function, is **baton passing**.

The token passing access method of Cambridge Ring ensures a very secure transmission network. As the calling DTE receives back its transmitted data, marked as accepted by the called DTE, it is *sure* the data has been received correctly. An error would be indicated if the accepted mini-packet was received back corrupted. A finite transmission delay between calling and called DTE can be guaranteed with a Cambridge Ring and so it may be used in real time applications. Collisions never occur because data flow is in only one direction.

## Transmission medium

Three main types of transmission media are commonly used to form LANs: twisted pairs of wire; coaxial cable; and optical fibre (although all other types may be adapted).

For bus topologies, coaxial cable is the most appropriate since it supports high data signalling rates over long distances. However, coaxial cable may be easily tapped – an advantage in the fact that the network may be easily modified, but a disadvantage in terms of security.

Optical fibre is suitable for both bus and ring networks. Its extremely wide bandwidths and consequent high data signalling rates mean that many network services may be multiplexed onto a single physical fibre network. Although a fibre network is not so easily adapted, it is difficult to tap and thus secure.

Twisted pairs of wires are only really suitable for use with ring networks (the Cambridge Ring was designed using twisted pairs), as each repeater in the ring regenerates all transmitted data, overcoming the bandwidth limitations of the wire.

## Glossary

| | |
|---|---|
| **access method** | the protocols and procedures to be followed to enable transmission of data on a local area network |
| **bus interface units** | active interfaces between DTE and a bus local area network |
| **Cambridge Ring** | ring local area network, using a token passing access method |
| **carrier sense multiple access with collision detect (CSMA/CD)** | access method on an Ethernet local area network. A bus interface unit wishing to transmit must first wait until the bus is free. If collisions of data do occur, all bus interface units cease transmission for random times |
| **collision** | presence of more than one data packet on a bus local area network |
| **contention** | where a number of bus interface units connect to a single bus, they must *contend* with each other for access to the bus |
| **distribution control** | a non-centralised control system, where a number of devices each maintain control, e.g. bus interface units on a bus local area network. Has the advantage that failure of one device will not stop system operation |
| **Ethernet** | bus local area network, using CSMA/CD access method |
| **mini-packets** | data packets, constantly transmitted around a Cambridge Ring local area network |
| **repeaters, ring interface units (RIU)** | DTE access points onto a ring local area network, performing a regenerative function for data transmitted around the ring |
| **token passing, baton passing** | descriptive terms for the access method commonly used by ring local area networks, in which packets of data are passed around the ring. If a packet is empty, a repeater may insert data for transmission |
| **topology** | general outline, in physical or logical terms, of a local area network |

14

MICROPROCESSORS

# 32-bit microprocessors

## A bit of a problem

In recent *Microprocessors* chapters, we have looked quite closely at examples of 4-bit, 8-bit and 16-bit microprocessors: a fictitious 4-bit microprocessor called SAM; the '8080 8-bit microprocessor; and the '9900 16-bit microprocessor. Throughout these chapters, however, we have never actually defined just what it is which makes one microprocessor, say, a 16-bit microprocessor, and another a 4-bit microprocessor.

This has been no omission, as it is difficult to make a straightforward distinction between, say, a 4-bit and an 8-bit microprocessor. In this chapter, however, we will attempt to clarify the situation by comparing the microprocessors we have already seen, along with some new ones. Two of the new microprocessors we shall look at are 32-bit microprocessors.

No problem would exist if all microprocessor architectures, i.e. internal layouts, followed a common pattern. *Figure 1* illustrates the architecture of a general purpose microprocessor which may be taken as our 'common pattern'.

This example microprocessor is an n-bit microprocessor. It has all the impor-

**1. Internal architecture** of a general purpose microprocessor.

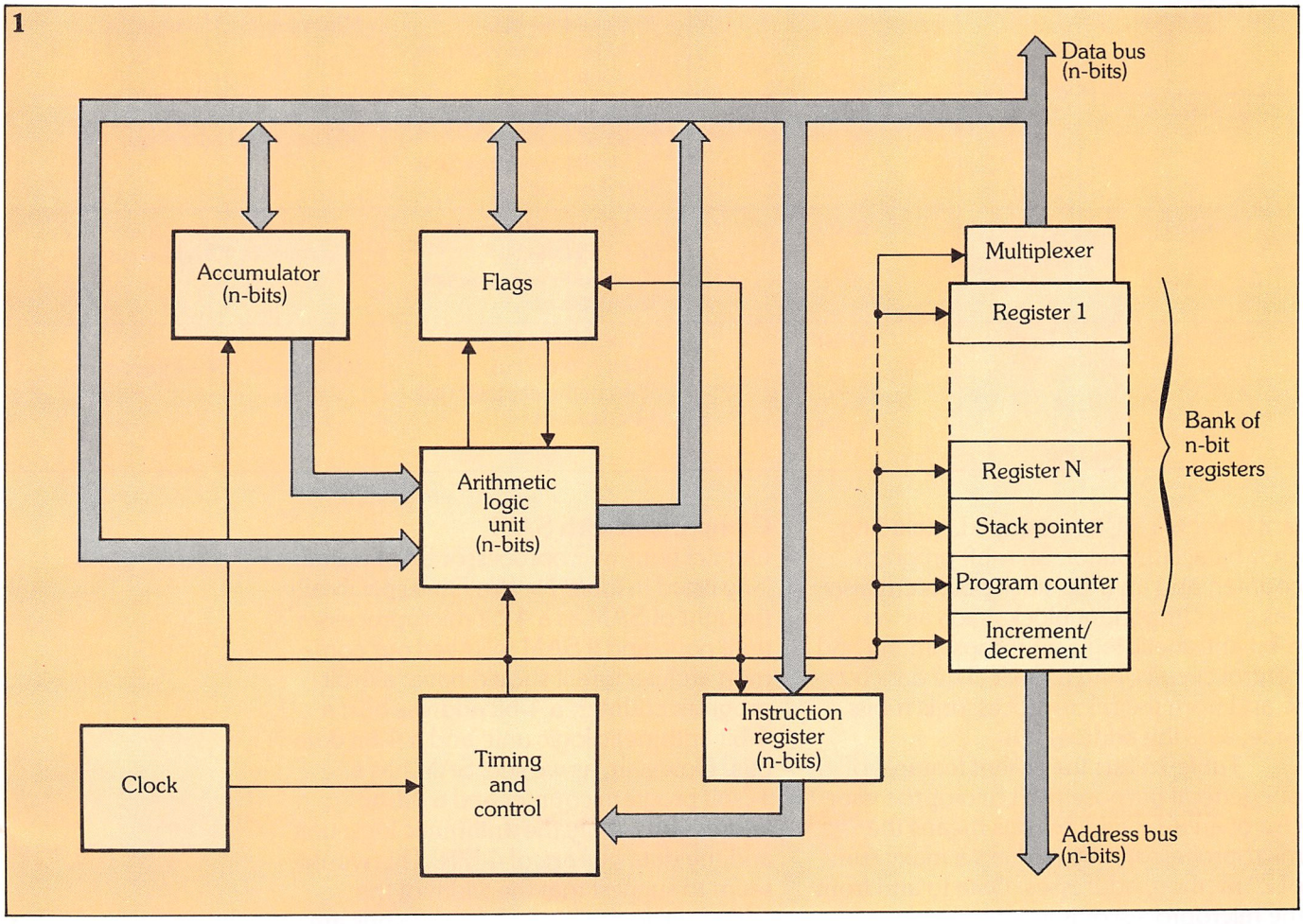


Table 1
**Summary of microprocessor features**

| Microprocessor | Bit classification | Features | Comments |
|---|---|---|---|
| **a)** General purpose microprocessor | n-bit | n-bit program counter<br>n-bit address bus<br>n-bit arithmetic logic unit<br>n-bit data bus | Maximum of $2^n$ addresses |
| **b)** SAM | 4-bit | 12-bit program counter<br>12-bit address bus<br>4-bit arithmetic logic unit<br>4-bit data bus | Maximum of $2^{12}$ addresses |
| **c)** '8080 | 8-bit | 16-bit program counter<br>16-bit address bus<br>8-bit arithmetic logic unit<br>8-bit data bus | Maximum of 64K addresses |
| **d)** '9900 | 16-bit | 16-bit program counter<br>15-bit address bus<br>16-bit arithmetic logic unit<br>16-bit data bus | Maximum of 32K addresses |
| **e)** '9980 | 16-bit | 16-bit program counter<br>14-bit address bus<br>16-bit arithmetic logic unit<br>8-bit data bus | Maximum of 16K addresses |
| **f)** '8086 | 16-bit | 16-bit program counter<br>20-bit address bus<br>16-bit arithmetic logic unit<br>8-bit data bus | Maximum of 1M addresses |
| **g)** '8088 | 16-bit | 16-bit program counter<br>20-bit address bus<br>16-bit arithmetic logic unit<br>8-bit data bus | Maximum 1M addresses |
| **h)** '68020 | 32-bit | 32-bit program counter<br>32-bit address bus<br>32-bit arithmetic logic unit<br>32-bit data bus | Maximum of 4G addresses |
| **i)** '68008 | 32-bit | 32-bit program counter<br>20-bit address bus<br>32-bit arithmetic logic unit<br>8-bit data bus | Maximum of 1M addresses |

tant internal registers required, including: an n-bit accumulator; an n-bit program counter; and an n-bit stack pointer. It also has other important blocks such as: an n-bit arithmetic logic unit; flags; timing and control circuits; and a clock. The data bus to and from the microprocessor is n-bits wide; as is the address bus.

*Table 1a* lists the salient features of our general purpose n-bit microprocessor. Use of an n-bit address bus means the microprocessor may address a maximum of $2^n$ memory addresses. Data to and from the microprocessor is in n-bit words.

## Comparison with SAM

Our fictitious microprocessor SAM is similarly listed in *table 1b*. We have specifically thought of SAM as a 4-bit microprocessor until now, and if SAM followed our common architecture it should have: a 4-bit program counter; a 4-bit address bus; a 4-bit arithmetic logic unit; and a 4-bit data bus. However, as we see, SAM has a 12-bit program counter and a 12-bit address bus. Only the arithmetic logic unit and the data bus are of 4-bits. This would seem to suggest that the width of the arithmetic logic unit and the data bus are

Science Photo Library/Jerry Mason

**Above:** this Commodore PET is being used in a pharmacy. It can handle stock control and store 'recipes' for pharmaceutical preparations.

the factors which define the bit classification of the microprocessor.

This theory is supported by the main features of another common microprocessor, the '8080 (*table 1c*). The '8080 is an 8-bit microprocessor with an 8-bit arithmetic logic unit and 8-bit data bus. Its program counter and its address bus are, however, 16-bit.

The features of the '9900 are listed in *table 1d* and we can see that it is a 16-bit microprocessor with a 16-bit arithmetic logic unit and a 16-bit data bus. Its program counter is also of 16-bits, but the address bus is only 15-bits wide.

A further microprocessor in the '9900 family is the '9980 – the microprocessor used in the TM 990/U89 microcomputer module. Its main features are listed in *table 1e* where we can see that it has a 16-bit program counter and a 16-bit arithmetic logic unit. In these respects it is similar to the '9900. But the '9980 only has a 14-bit address bus which is capable of addressing only $2^{14}$ = 16K memory addresses. Furthermore it only has an 8-bit data bus!

A similar situation arises with the architectures of two similar microprocessors of a different device family: the '8086 and the '8088, whose features are listed in *table 1f* and *1g* respectively. The '8086 is obviously a 16-bit microprocessor as it has a 16-bit arithmetic logic unit and a 16-bit data bus. It features a 16-bit program counter and a 20-bit address bus, too. The '8088, on the other hand, has identical features – apart from only having an 8-bit data bus.

So, the question arises, are the '9980 and the '8088 microprocessors *really* 16-bit microprocessors at all? If they can only acquire data in blocks of 8-bits at a time, are they not 8-bit microprocessors?

As it takes longer for these microprocessors to acquire data in 8-bit blocks than it does for their true 16-bit counterparts, then overall times to perform the same operations must be correspondingly greater – between about 10% and 40%, depending on the particular operations, as it happens.

Nevertheless, the arithmetic logic unit (including the accumulator) of both microprocessors is of 16-bit construction. For this reason, we must adapt our earlier theory of microprocessor bit classification accordingly: that the bit classification of a microprocessor is defined by the bit-length of the processing blocks – the arithmetic logic unit and the accumulator. This is irrespective of the data bus width, the address bus width, the program counter, instruction register, stack pointer etc.

Two new microprocessors are listed in *tables 1h* and *1i*. The '68020 is a true 32-bit microprocessor in which all features are of 32-bits. The '68008, on the other hand, is a derivative of the '68020, having only a 20-bit address bus and an 8-bit data bus. The '68020 microprocessor has recently gained a great deal of critical attention as the microprocessor used in the new Sinclair QL computer.

*Figure 2* illustrates the pin assignments and general package size of the '68008 microprocessor. *Figure 3* shows the main input and output signals of the microprocessor, organised into groups for easier reference. The following sections describe the important inputs and outputs, in a clockwise direction, starting from the top right.

### Address bus
Outputs A0 to A19 form a 20-bit three-state address bus which provides a memory address at all times except during interrupt acknowledge cycles. During such cycles, outputs A1, A2 and A3 provide information about the level of interrupt that is being serviced. Outputs A0, and A4 to A19 are all high. The 2-bit address bus allows a directly addressed memory of up to 1 Mbyte to be accessed by the microprocessor.

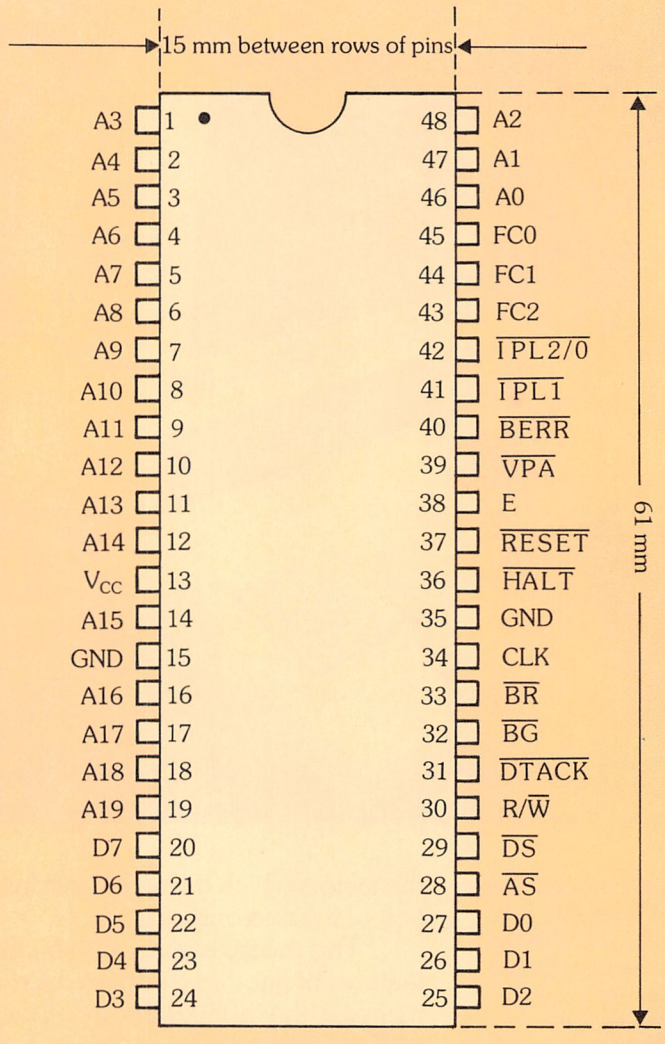
### Data bus
Lines D0 to D7 form a bidirectional, three-state data bus to input or output data.

### Asynchronous bus control
Asynchronous data transfers are controlled with the following signals:
1) **address strobe** ($\overline{AS}$) – a three-state ouput signal which indicates that there is a valid address on the address bus;
2) **read/write** (R/$\overline{W}$) – defines whether data on the data bus is for a read operation (i.e. input from memory) or a write operation (i.e. output to memory);
3) **data strobe** ($\overline{DS}$) – a three-state output signal which indicates there is valid data on the data bus;
4) **data transfer acknowledge** ($\overline{DTACK}$) – input, indicating to the microprocessor that an input or output data transfer is finished.

### Bus arbitration control
Two lines are used to determine the bus master device if two or more microprocessors are connected in one system. These lines are:
1) **bus request** ($\overline{BR}$) – all connected devices are wire-ORed with this input. An input on this line indicates to the present bus master device that another device wishes to become bus master;
2) **bus grant** ($\overline{BG}$) – an output on this line indicates to all other potential bus master devices that the microprocessor will release bus control at the end of the current bus cycle. The device which requested control, as in (1) above, becomes next bus master.

### Interrupt control
Two input lines ($\overline{IPL0}/\overline{2}$ and $\overline{IPL1}$) are used by a device requesting an interrupt, to indicate that interrupt's priority.

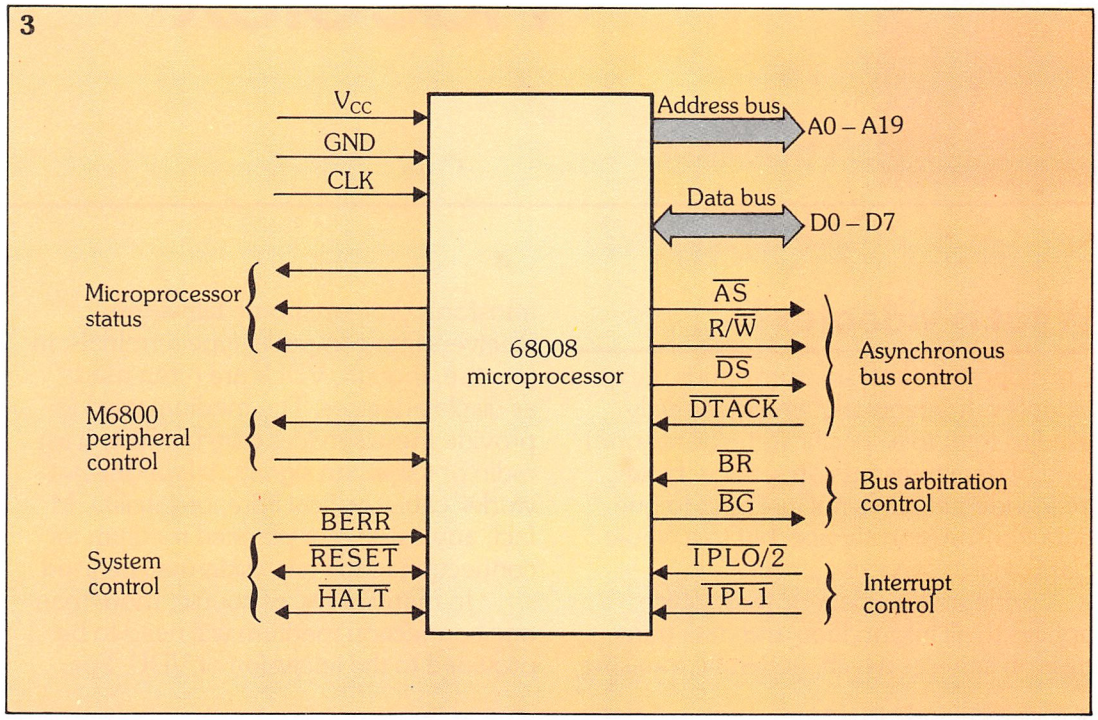
2



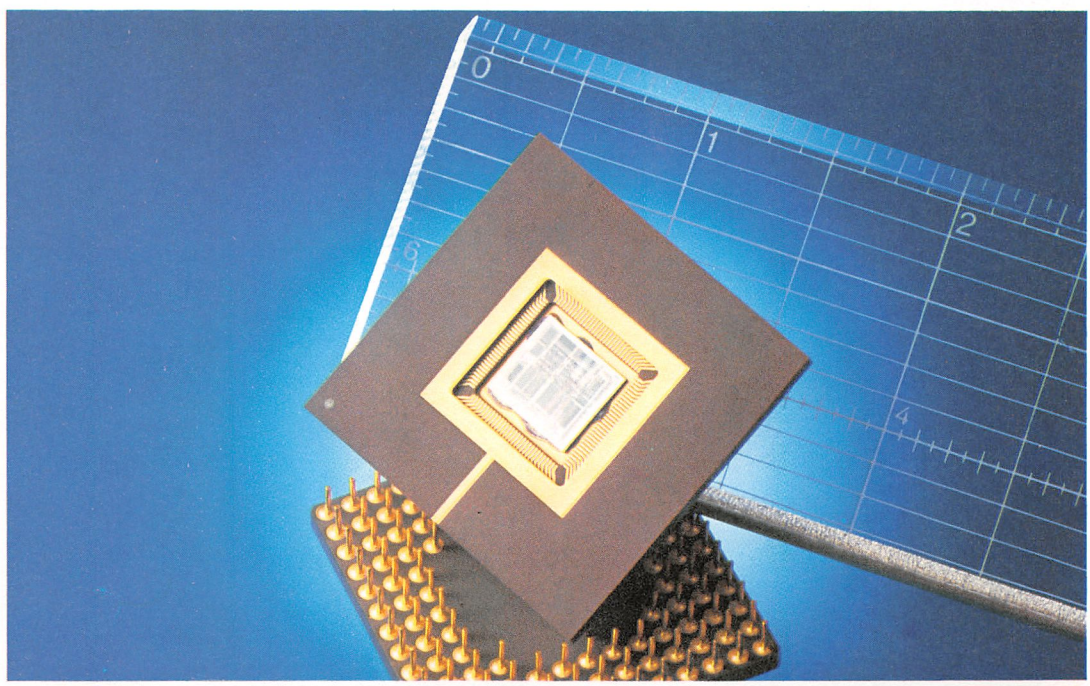

**2. Pin assignments** for the '68008 microprocessor.

### System control
Three lines are used to reset or halt the microprocessor, or to indicate error:
1) **Halt** ($\overline{HALT}$) – an input on this bidirectional control line causes the microprocessor to stop at the completion of the bus cycle currently being processed. When the microprocessor has stopped executing instructions, it signals an output on this line.
2) **Reset** ($\overline{RESET}$) – an input on this bidirectional line resets the microprocessor. An output from the microprocessor on this line resets all external devices.
3) **Bus error** ($\overline{BERR}$) – an input on this control line indicates to the microprocessor that there is a bus error. If bus error and halt signals are input simultaneously to the

**3. The '68008's main** input and output signals.




**Right:** the MC68020 microprocessor. (Photo: Motorola).



microprocessor, the cycle currently being executed will be re-run.

**6800 peripheral control** – allows the '68008 (an asynchronous device) to be interfaced with peripheral devices in the synchronous '6800 family.

### Microprocessor status

Three ouputs indicate the type of cycle currently being processed. There are eight different cycle types.

### Clock

A constant frequency clock signal is required by the microprocessor, at this input.

### Power

Power is supplied to the microprocessor on these lines: $V_{CC}$ is 5 VDC and GND is 0 VDC.

21

COMMUNICATIONS

# Videotex

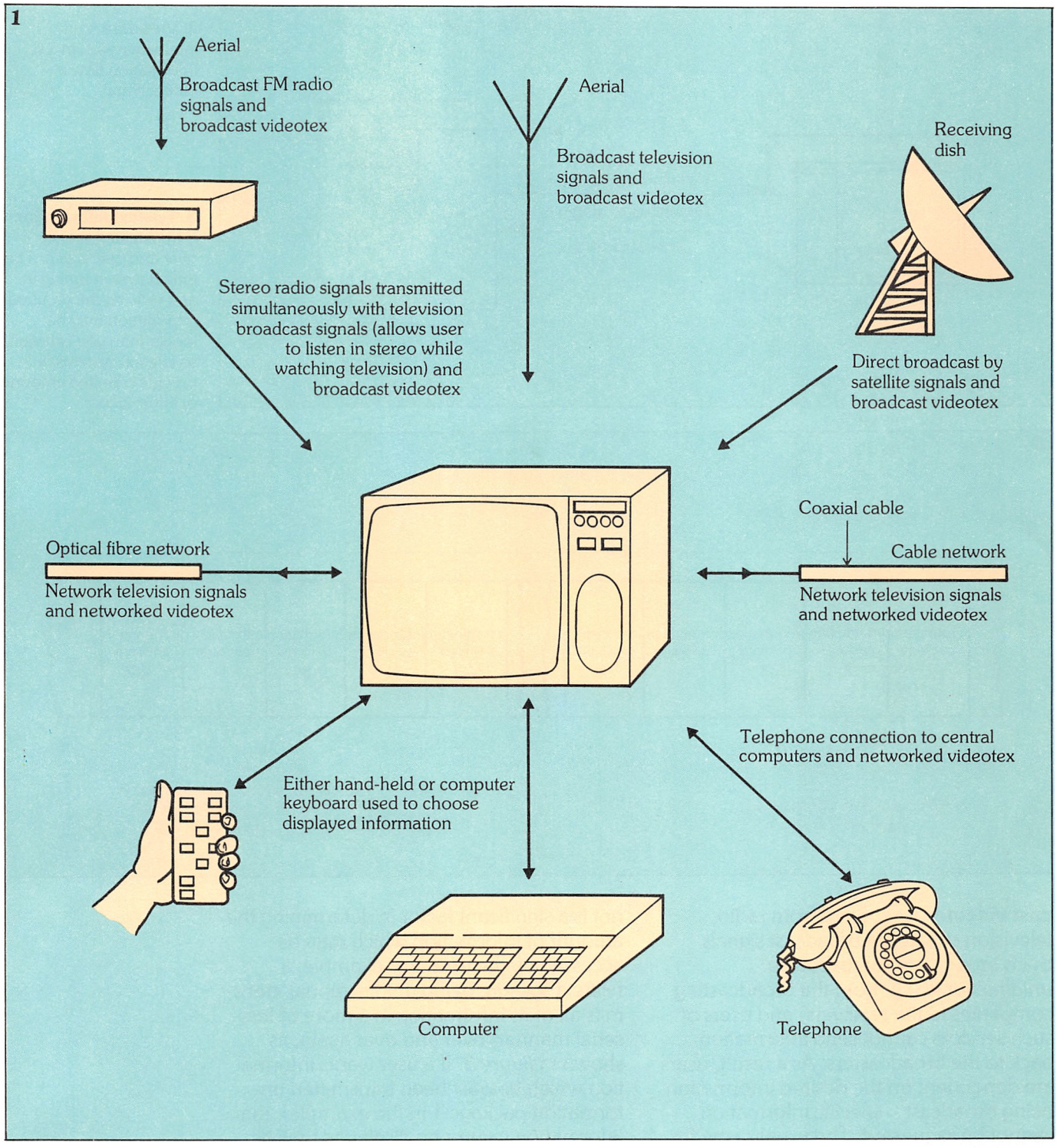
## What is videotex?

Considerable confusion surrounds the term *videotex*. It is regularly, and incorrectly, used to refer to a specific (often, only one) type of data transmission service. However, videotex does not just refer to one data transmission service, but to a whole *class* of such services.

Videotex is a generic term which applies to all information and data transmission services which, generally speaking, transform ordinary home television receivers into computer-type terminals; in practise, though, VDUs are often used as display devices. The medium used to provide these services may be: broadcast radio or television signals; telephone networks; cable; optical fibre; or satellite. In fact, any existing or planned medium for connection to the television may be used.

In many cases, of course, more than one connection medium will need to be provided to the teleivsion or VDU. For

**Below:** travel agents such as this one are members of private viewdata groups on the national Prestel network. These groups are known as closed user groups because the public is denied access to the information.
(Photo: British Telecom).

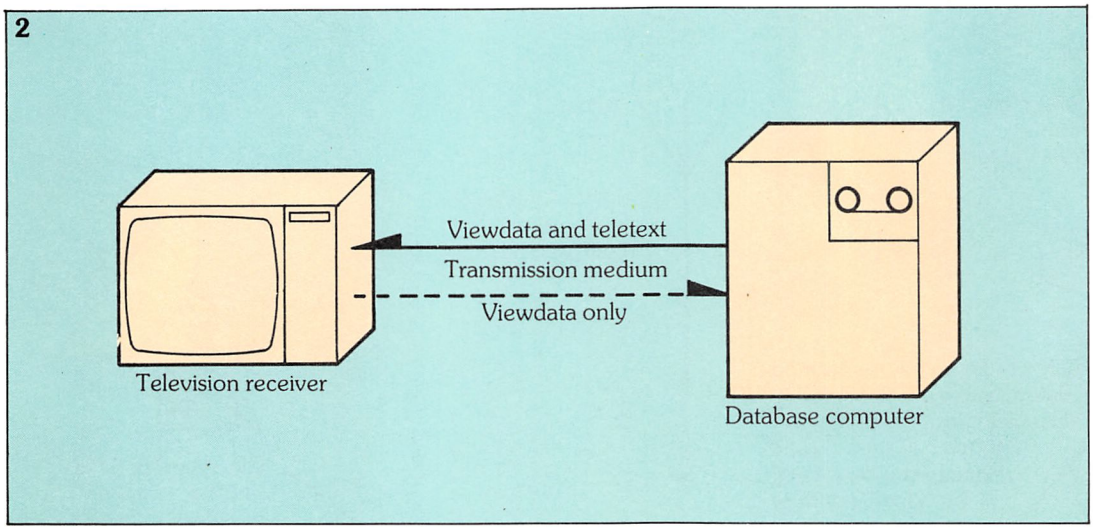



**1. The home television receiver** or VDU will soon be able to receive all forms of videotex services.

example, a television receiver receives broadcast television signals via a receiving aerial and downlead – it must also be connected to whichever medium the information service is provided on, for example via a telephone connection point for access to the telephone network.

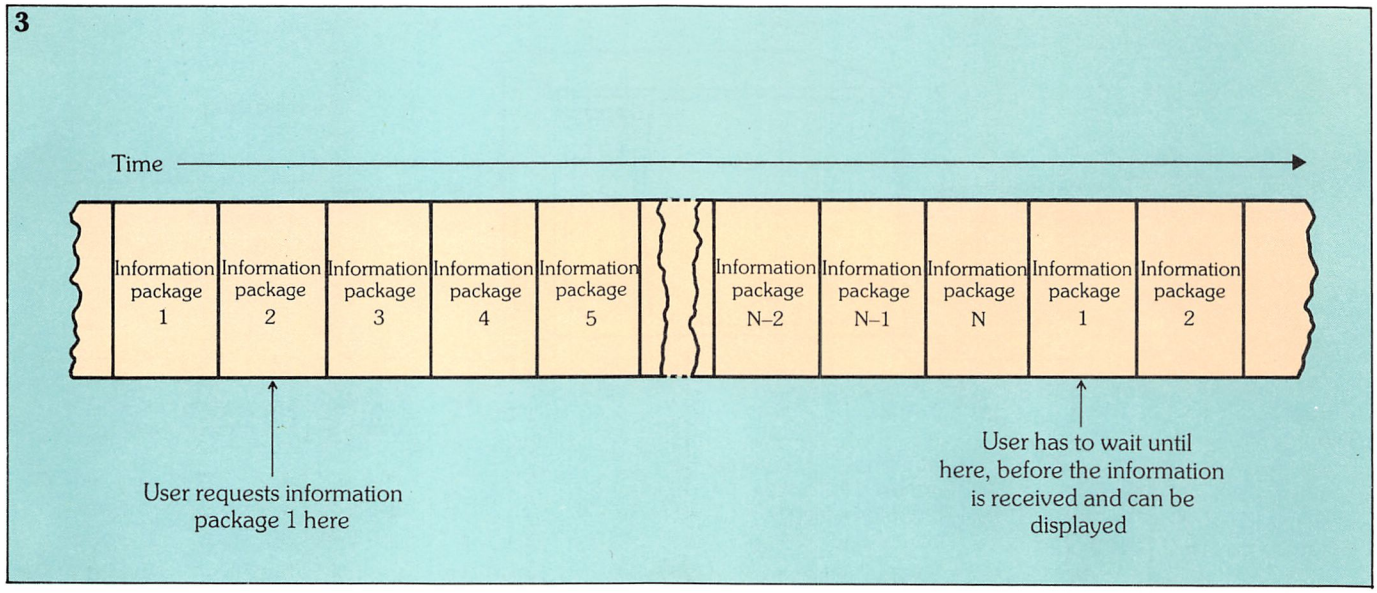
Indeed, it is reasonable to assume that in the not too distant future, the home television receiver or VDU will be connected to all possible media and be capable of receiving *all* forms of videotex services. *Figure 1* illustrates an example.

From this we can see that there are two main forms of videotex: broadcast videotex; and networked videotex. **Broad-**




**2. Viewdata** is an interactive system, i.e. information flow is bidirectional.

**3. Teletext information** is transmitted serially – over and over again. The time between tuning in and viewing the required information may be longer than you can wait, so teletext systems only provide a limited amount of information.

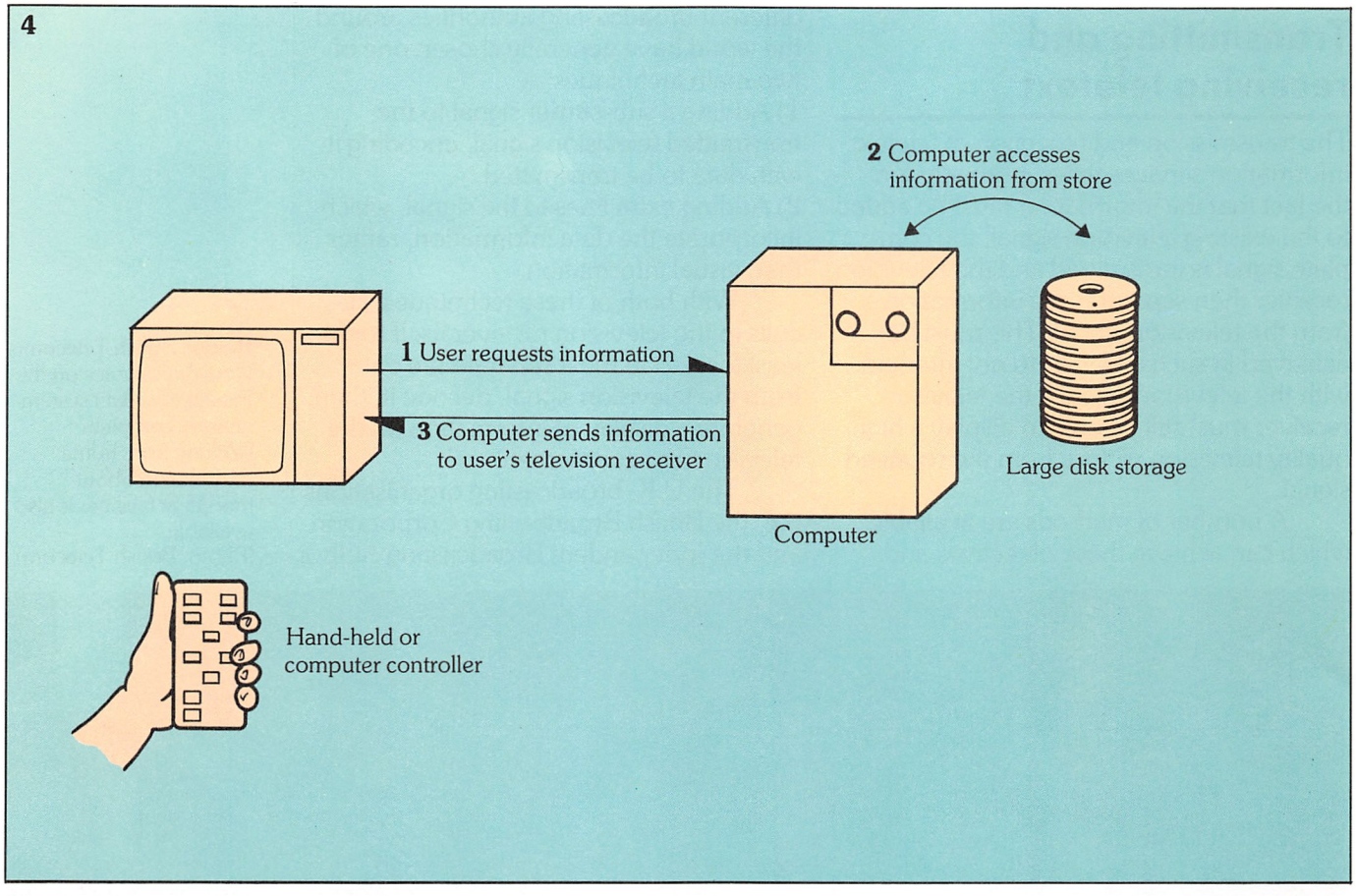



**cast videotex** is received from radio, television or satellite broadcast signals. Such broadcasts are, of course, *unidirectional* (i.e. from the broadcasting companies to the receivers) and users of such services cannot send information back to the broadcasters. As a result, users are dependent on the desired information being broadcast – specific information cannot be requested. The usual name for broadcast videotex is **teletext**.

**Networked videotex**, on the other hand, features *bidirectional* information flow, i.e. it is an **interactive** system. Users are able to request specific information and fairly quickly have access to it. The usual name for networked videotex is **viewdata** (*figure 2*).

This feature of being interactive or not is a significant factor in determining the amount of information which may be accessed by the user. For example, a teletext system, being unidirectional, transmits all of its information in a more or less serial manner, over and over again, as shown in *figure 3*. If a user wants information which has just been transmitted (information package 1 in the example), that information cannot be displayed until its turn to be retransmitted comes round again. The length of time between the user tuning in and the eventual display of the information wanted may be longer than the user is prepared to wait! Because of this, non-interactive videotex services, such as teletext, typically provide only *small* quantities of information.

Interactive videotex such as viewdata,




**4. Considerably more information** is available to the viewdata user as a central computer database can be accessed on demand.

on the other hand, does not suffer from this disadvantage. User's may request specific information directly from the information provider (generally a computer with large storage facilities) which then rapidly accesses the requested information from store and transmits it to the user's television or VDU to be displayed (*figure 4*). The amount of information accessible by the viewdata user via the computer is thus much greater than that accessible by the teletext user.

**The development of videotex services**

Teletext, being generally a broadcast videotex service, has been developed primarily by the broadcasting organisations, wishing to provide customers with an extra feature without the need for expensive, complicated equipment.

Viewdata, on the other hand, was, until recently, developed by the PTTs (postal, telegraph and telecommunications authorities), for use over the telephone network. It would seem, therefore, that little or no standardisation between the two types of videotex service would have emerged from this largely bilateral development. However, as the PTTs pitched their sales drive at the *private* consumer (instead of the business user – which was where a system such as this was needed, as it turned out), it was assumed that potential users would wish to adapt an existing display device, i.e. a television receiver, rather than buy a VDU.

This mistaken marketing ploy has proved advantageous for users as essentially similar standards have been developed for both types of videotex. Thus, one television receiver, or VDU, may be used for the reception and display of both.

Recent development in both viewdata and teletext services have shifted emphasis away from the private customer service towards the business user – with one or two notable exceptions. However, the videotex standards already exist and are difficult to change, so even purely business videotex services are essentially compatible with those to which the private user has access.

## Transmitting and receiving teletext

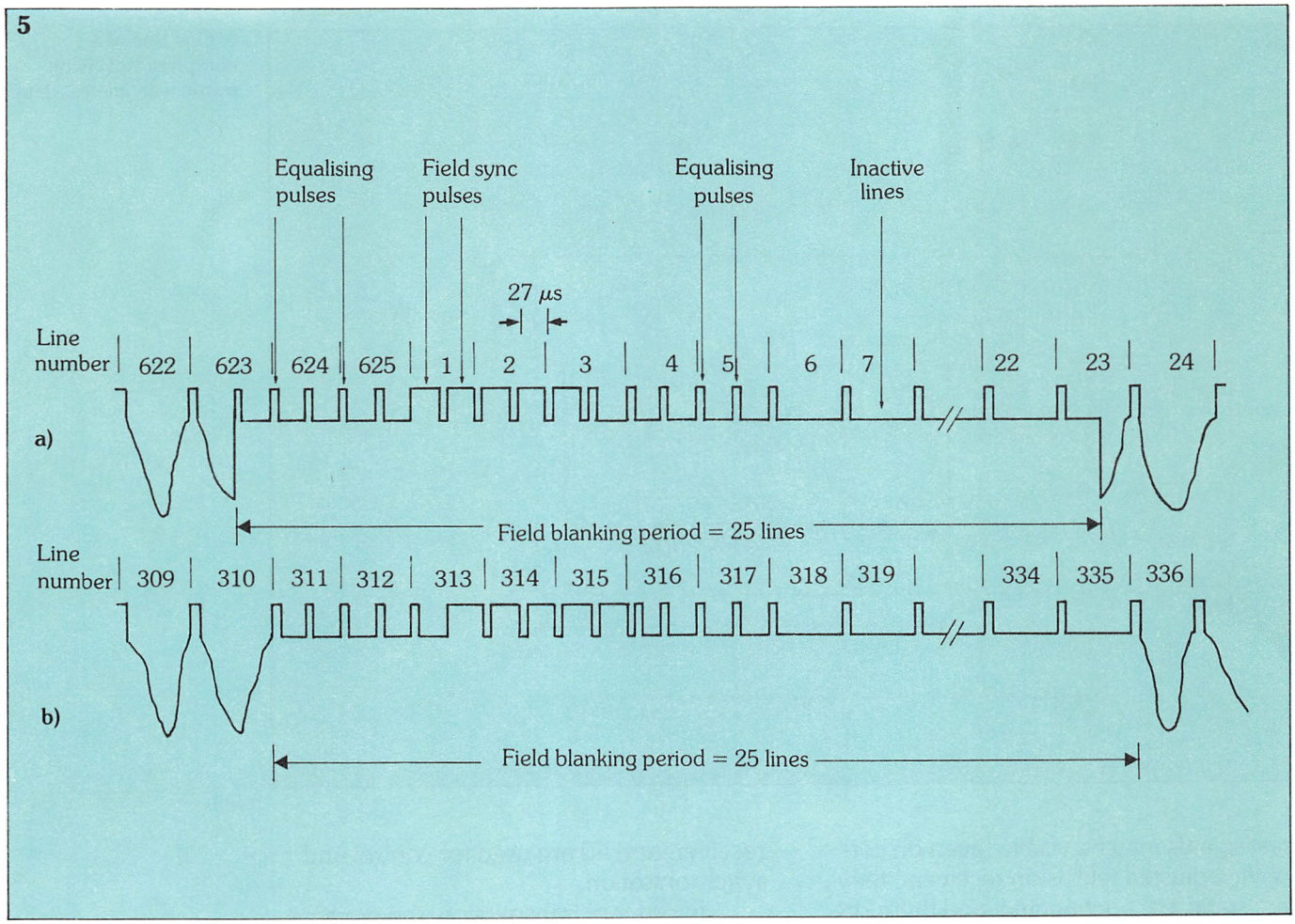
The transmission and reception of teletext information services relies, primarily, on the fact that the information must be added to the existing television signal; the complete signal is transmitted and the television receiver then separates the information from the television signal. This must be achieved in such a way as to not interfere with the television signal – the televison receiver must still be able to display a high quality television picture from the received signal.

A number of methods are available which can achieve these objectives and different broadcasting authorities around the world have generally chosen one of two main techniques:
1) Adding a sub-carrier signal to the transmitted television signal, encoding it with data to be transmitted.
2) Adding extra lines to the signal, which incorporate the data information, rather than visual information.

With both of these techniques, circuits in the television receiver itself (or at least local to it) must separate the data from the television signal, decode it, then generate a display of information on the television receiver screen.

The U.K. broadcasting organisations (i.e. the British Broadcasting Corporation and the Independent Broadcasting Author-


**Below:** British Telecom's Homelink service on the Prestel viewdata system enables complete banking from home. Information about mortgage facilities is also available.
(Photo: British Telecom)




**5. (a) A composite monochromatic video** signal occurring between an odd and an even field; **(b)** the signal between an even and an odd field.

ity) agreed on a standard based on the second of the two techniques – adding extra lines of data – in October 1974.

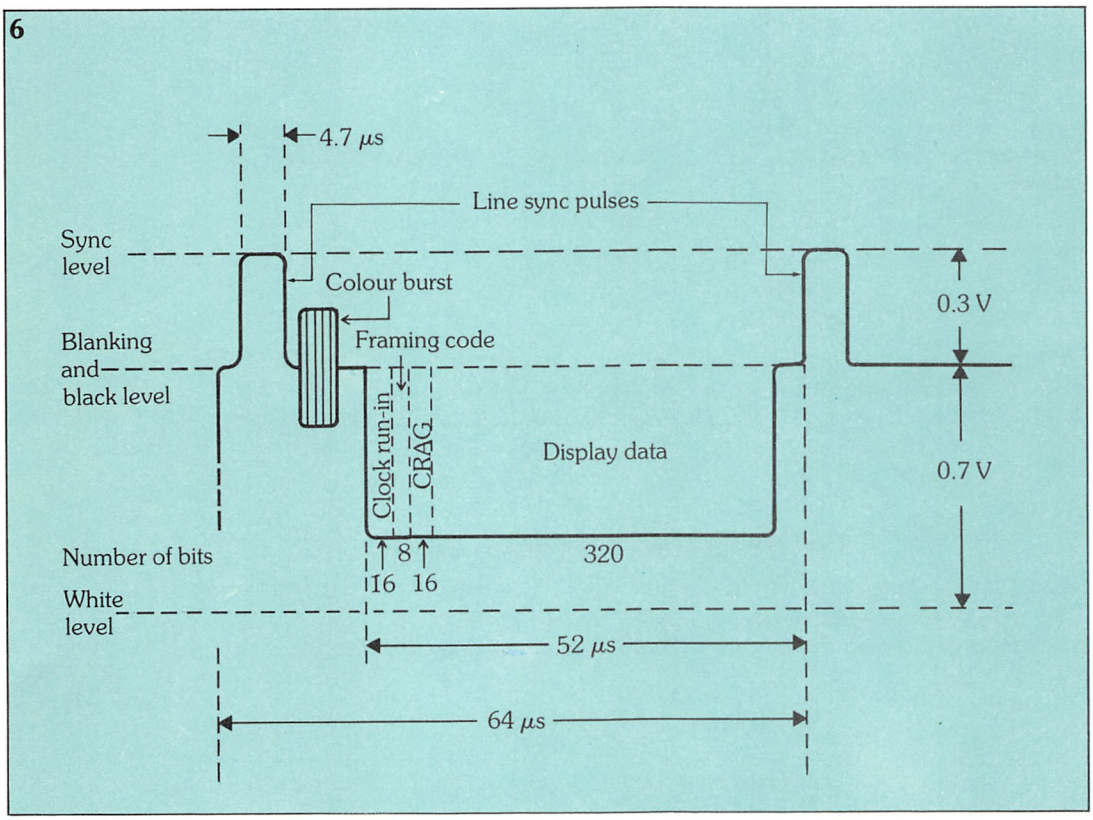
In practise, no *extra* lines are actually added – instead previously unused lines are used. To appreciate how this is done, we'll look again at the composite television signal waveform considered in *Communications 4*.

*Figure 5* illustrates a possible composite video signal occurring between the odd and even fields of a complete television picture. The field blanking periods (i.e. the periods between fields, during which the electron beam is blanked so no display appears on the television screen) are of 25 lines between each field. A total of 50 out of 625 lines for each frame of the displayed television picture are therefore blanked.

Some of these blanked lines are used to hold field synchronising pulses and equalising pulses, but a considerable number of them (lines 6 to 22, and 318 to 335) are inactive. Teletext transmissions by the U.K. broadcasting organisations take advantage of these inactive lines, by transmitting digitally encoded data on them.

Not *all* of these inactive lines may be used, however, and the choice of those most suitable is defined by two major factors. First, the field flyback period of some receivers (particularly old ones) is quite long, continuing well past the field synchronisation and equalisation pulses of the field blanking periods, into the series of inactive lines. If information is transmitted in the early lines of the series, there is a chance that it may be displayed diagonally up the screen, as the field flyback occurs.

The second factor determining the choice of inactive lines for teletext transmission is that data on lines late in the series may appear at the top of the screen, if the television receiver's picture height is incorrectly adjusted. Some of these later inactive lines are also used occasionally for signals, inserted by the broadcasting organisations for test purposes. Although these




**6. One line of a complete television frame** with teletext data.

test signals may be visible (even on correctly adjusted television receivers) they are static, i.e. unchanging, and therefore not annoying to the viewer.

Teletext data lines, on the other hand, comprise patterns of binary data, producing light and dark variations of dots if seen on the television screen. As each line is serially transmitted, the patterns of light and dark dots correspondingly change, producing an annoying display to the viewer.

The lines used for the transmission of teletext are therefore chosen to be approximately central in the series. When the teletext services (**Ceefax** – BBC; **Oracle** – IBA) first began in 1974, lines 17 and 18 of the blanking period between odd and even fields, together with lines 330 and 331 between the even and odd fields, were used. Since then, other lines have been included for teletext transmission purposes but the principle remains the same.

One line of the complete television frame with teletext data is shown in *figure 6*. A total of 360 bits are transmitted during the line, of which 320 are data bits to be decoded and displayed by the televison receiver, and 40 are used for control and synchronisation.

In order of transmission, the purpose of these bits is as follows:

**1)** 16 bits (2 bytes), known as **clock run-in** bytes. These have a similar function to the colour burst signal of a transmitted television line of a colour picture – to synchronise a reference oscillator within the receiver. This is achieved simply by transmitting, during both bytes, a continuous stream of bits alternating between 1 and 0. The clock run-in has a secondary function in that the frequency of occurrence of the signal is a specific frequency, which is detected at the television receiver and used to identify the line as a teletext transmission and not another, e.g. a test signal.

The frequency of occurrence may be calculated, as we know that 360 bits of data are transmitted in 52 μs, so one complete cycle period of alternate 1 and 0 bits is:

$$\frac{2}{360} \times 52 = 0.289\ \mu s$$

The frequency of the signal with this period is therefore:

$$\frac{1}{0.289 \times 10^{-6}} = 3.46 \text{ MHz}$$

**2)** 8 bits (one byte) known as the **framing code**. The eight bits of this byte are always the same: 11100100, and indicate that the following bytes (a total of 42) are all to be taken as data.

**3)** 2 bytes of data known as the **control and row address group** (CRAG). Row addressing refers to the **rows** of characters generated on the screen of the television receiver. The display data which follows must be identified as belonging to one particular row of one particular **page**, i.e. screenful, of information. Further, each page must be identified as belonging to one particular collection of pages known as a **magazine**.

The CRAG bytes first identify the magazine, and second the row which the following display data belongs to. (It is left to the display data itself to identify the page number!) The two CRAG bytes form a 16-bit group which comprises a 3-bit group which identifies the magazine number (1 to 8), and a 5-bit group to identify the row number (0 to 31 – although only 24 rows, i.e. 0 to 23, are displayed). The least significant bits of each group are transmitted first.

These two groups are encoded before transmission to enable an error detection and correction process to be carried out at the television receiver. The code used (**Hamming code**) doubles the number of bits which must be transmitted in the CRAG bytes to 16.

**4)** The remaining 40 bytes of data (i.e. 320 bits) are purely display data, all 40 bytes corresponding to characters and graphical symbols to be displayed on the television receiver screen. These characters and symbols are encoded into bytes, roughly following ASCII code.

*Figure 7* illustrates a possible line of teletext, complete with information for display.

### Requesting information display

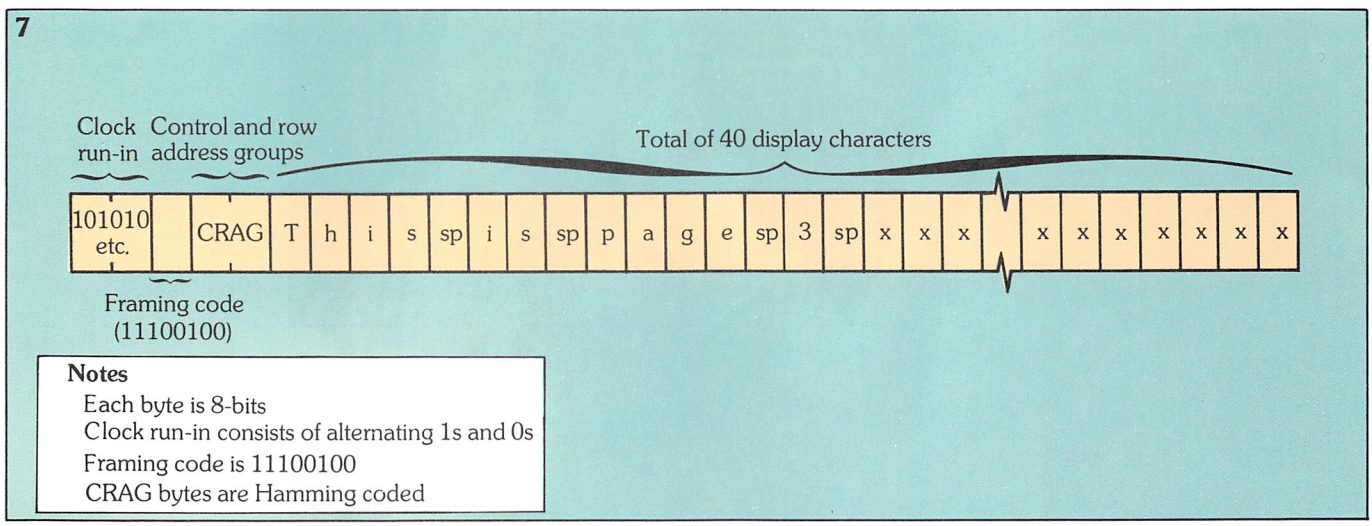
We can now take a further look at the way delay between information request and its eventual display can determine the amount of information accessible to the user of a non-interactive videotex service.

There are 40 characters in one row of displayed information, so one transmitted teletext line corresponds exactly with one displayed row. This means that 24 transmitted teletext lines are transmitted for every full page of information.

Using the system we have outlined here, where four teletext lines are transmitted per complete television frame and 25 frames are transmitted every second, this means that just over four pages of information may be transmitted each second.

If we chose an arbitrary time period, for which a user may be prepared to wait until the requested information is displayed – say, 10 seconds – the maximum number of pages allowed in one magazine may be calculated to be about 40. This is a complete total of about:

$40 \times 40 \times 24 = 38{,}400$ characters




**7. An example line** of teletext with data to be displayed.


**Left:** some information on Prestel is provided free to the subscriber. The information being paid for by the advertiser of the product or service. (Photo: British Telecom).

Not really a lot, considering the information might be broadcast to millions of viewers, each potentially requiring different types of information.

Fortunately, there are a number of techniques available for increasing the amount of information without increasing access time. First, as we mentioned earlier, more inactive lines may be and, in fact, are used for teletext transmission purposes. If the number of lines used is doubled, the access time may be halved; the number of pages of information available may be doubled, or some 'happy medium', trading off amount of information and access time, may be reached.

The second technique depends on the actual information itself. If pages of information are held with blank rows, i.e. with no row information to be displayed, then it is not essential to transmit them. Untransmitted lines cannot be received (because they're not there) so they will be displayed as blank, anyway.

Another way, which is really just a trick, merely gives the appearance that access time is shorter than it actually is. Those pages of information which are estimated by the broadcasting organisations to be most frequently used are transmitted more often than others. For example, the index pages of the magazines are transmitted very often, say, once every two seconds, allowing rapid access. Other pages, not so regularly transmitted, may take, say, 8 seconds to access.

Although other methods of reducing access times and increasing accessible information are both in use and under development, non-interactive videotex systms can *only have a limited amount of information.*

## Viewdata

Viewdata was developed by the PTTs as a means of increasing revenue from the PSTN. In the U.K. the main viewdata service, **Prestel**, is operated by British Telecom.

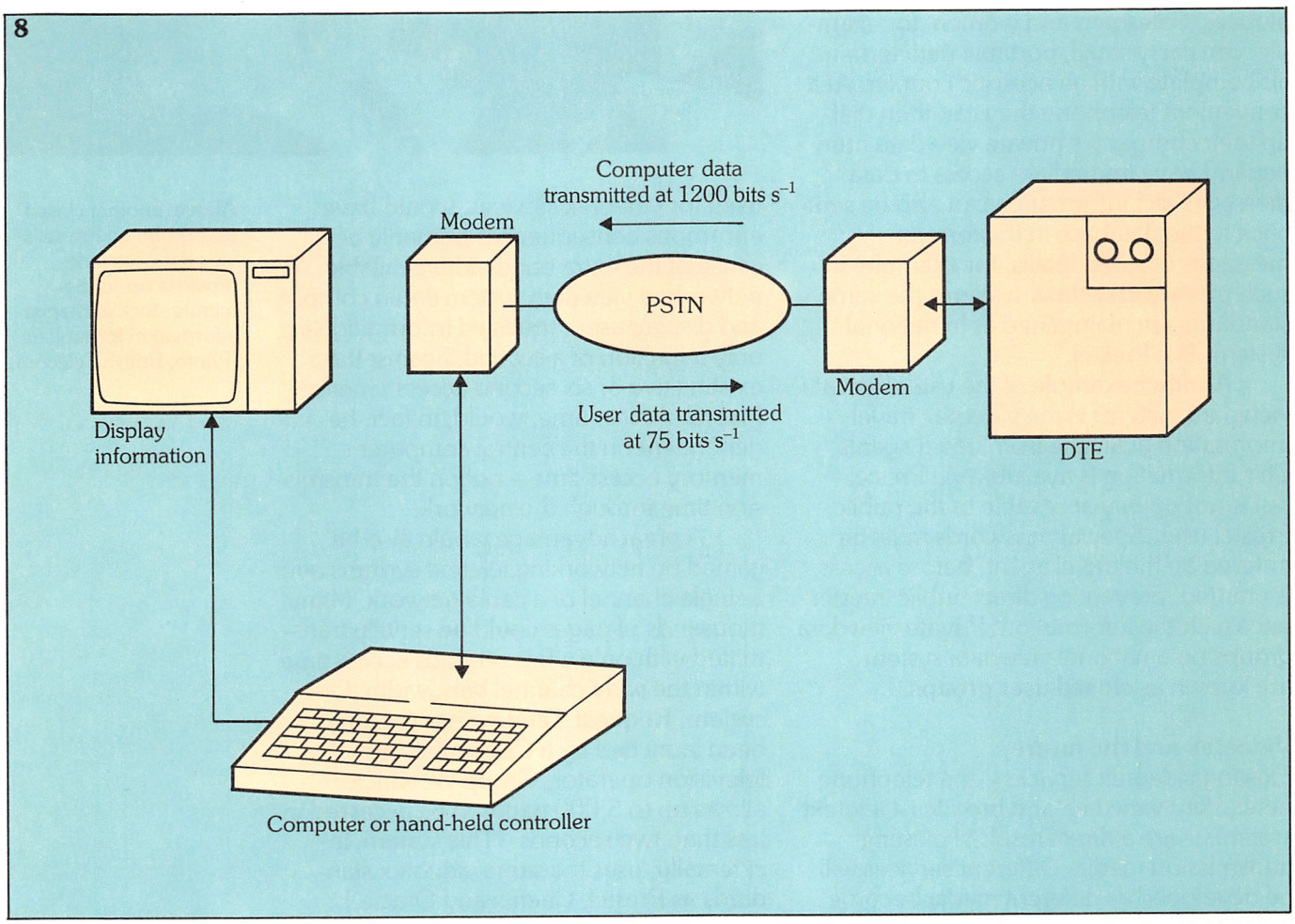
Viewdata systems are very similar to teletext systems in that the data transmitted from the central database to users' television receivers is organised in more or less the same way. The formats of viewdata and teletext lines are virtually identical, at least within any one country: Prestel, Ceefax *and* Oracle, for example, display 24 rows of information, each with 40 characters or graphical symbols, which form one page of information.

Prestel data is transmitted over the PSTN using standard modem-type communications, according to CCITT recommendation V.23. This recommendation, as illustrated in *figure 8*, defines the data signalling rate from the centre database computer to the user as 1200 bits $s^{-1}$. User data (sent to the computer) is transmitted at a data signalling rate of only 75 bits $s^{-1}$.

At first sight, it might appear that the computer-to-user data signalling rate is considerably lower than the equivalent rate for teletext transmission. A single line of information corresponding to one full displayed row has 360 bits and therefore takes about 0.3 second to be received. A full 24 row page thus takes about eight seconds to compile and display. However, like teletext, it is unnecessary to transmit blank rows, so any page with blank rows takes less time to compile. Further, if any row is not entirely full, say, it has only 30 characters, there is no need to transmit the last 10 characters, saving 25% of transmitted data.

Even with these savings, it may still take an average five or six seconds to compile and display a requested page, so

8




**8. CCITT recommendation V.23** governs the transmission of Prestel data over the PSTN.

little or no advantage is gained in time over existing teletext systems. There is, on the other hand, a distinct advantage in terms of the *amount* of accessible information – this being limited only be the size of the store. Prestel, for example, contains about 0.25 million pages of information and more are to be included.

### Private viewdata

Prestel is an example of the *public* use of viewdata. In many organisations there is a need to provide information that can be both requested and updated by employees. Salesmen and women, for example, can carry small, portable data terminals complete with an acoustic coupler. At a convenient telephone they can then dial up their company's private viewdata number and have immediate access to data, messages etc; information can also be sent back to the database in the form of messages or sales details, for example. In such **private viewdata** systems, the same standards are maintained as in national systems like Prestel.

Another example of the use of private viewdata systems is the videotex travel information available from travel agents. This information is available via Prestel, but is not openly accessible to the public Prestel user. Special passwords must be entered by the travel agent, before access is granted, preventing direct public interference with the information. Private viewdata groups on a national viewdata system are known as **closed user groups**.

### Videotex and the future

Existing videotex services – the telephone networked viewdata and broadcast teletext systems – are a direct result of existing transmission media. Different services will be developed as different media become more common.

Over the next few years there will be two main changes in the media available for television and telephone services, both of which will have consequent effects for videotex. The first change will be the incorporation into the public network of private cable television services using coaxial, or optical fibre, media for transmission.

A single cable television channel, if used for videotex services, would have enormous consequences primarily because of the extra bandwidth available. A networked viewdata system could compile and display user requested information in only a fraction of a second, against the existing five or six second access time for Prestel. Access time, would, in fact, be dependent on the central computer's memory access time – not on the transmission time through the network.

A great advantage would also be gained by networking teletext systems over a single channel of a cable network. Many thousands of pages could be serially transmitted with only a few seconds access time within the wide channel bandwidth. One system, **Request Teletext** has recently been launched by a Californian cable television operator, Group W, which allows up to 5,000 pages to be accessed in less than two seconds. (This system, incidentally, uses the same videotex standards as Prestel, Ceefax and Oracle.)

The second change in transmission media will be the incorporation of System X telephone exchanges and the integrated services digital network (ISDN) into the existing telephone system. This will provide digital data links, allowing data signalling rates of 64,000 bits $s^{-1}$ over telephone lines. With data signalling rates of this magnitude, access times on the existing Prestel system could be reduced so that the user has immediate response.

**Above:** another closed user group on Prestel's viewdata network provides up-to-the-minute stock exchange information for brokers. (Photo: British Telecom).

# Glossary

| | |
|---|---|
| **Ceefax** | the BBC's teletext service, a broadcast videotex information system |
| **clock run-in** | the first 16 bits of a teletext line, of alternate 1s and 0s, to synchronise the oscillator at the television receiver |
| **closed user groups** | restricted bodies of users to certain information on British Telecom's Prestel viewdata service |
| **control and row address group (CRAG)** | fourth and fifth bytes of a teletext line, containing information relating to the row number and magazine of the display data following it |
| **framing code** | a unique code in a teletext line, identifying the line as being a teletext transmission |
| **interactive** | a two-way information service, in which the user may request as well as receive information |
| **magazine** | a number of teletext pages |
| **Oracle** | the IBA's teletext service, a broadcast videotex information service |
| **page** | a displayed amount of videotex information, consisting of up to 24 rows of characters, with up to 40 characters per row |
| **Prestel** | British Telecom's networked videotex viewdata system. Around 0.25 million pages are accessible to users |
| **private viewdata** | networked videotex information system which is accessible only by selected users |
| **Request Teletext** | a recently developed networked teletext information system, in which 5,000 pages of information are accessible within two seconds |
| **row** | a line of up to 40 characters displayed on a videotex information system |
| **teletext** | a non-interactive videotex information system. Information in a teletext system is constantly transmitted more or less serially, over and over again |
| **videotex** | information systems which allow users to access information stored at a central database, generally via a home based television receiver |
| **viewdata** | an interactive videotex system. Information is only transmitted to the user when requested |

# The transputer

## More than a microprocessor

The **transputer** – an innovation and potentially great improvement in computer technology – has been developed by INMOS, the British semiconductor manufacturer set up in 1978. Although not yet in full production, the INMOS transputer is a contender in the fight over the design of the next generation of advanced 32-bit microprocessors.

The transputer's design philosophy, however, proves it to be different from a microprocessor. The transputer's name is derived from the words **trans**istor and com**puter**, and this sums up the intention of its designers for it to be a programmable silicon component.

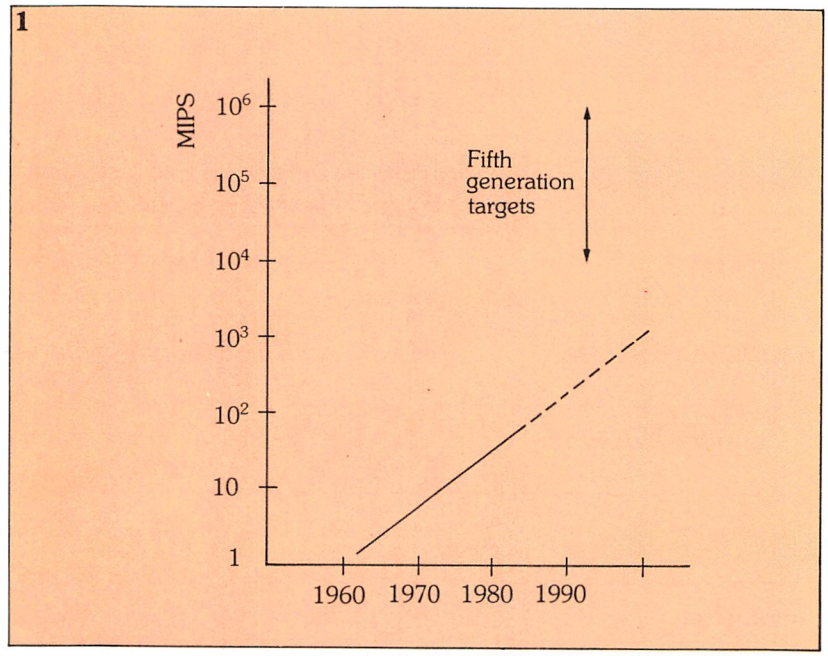
Computer system performance has increased by a factor of ten each decade. This continual improvement has been made possible by advances in circuit technology and the development of increasingly complex systems. However, although the VLSI (very large scale integration) technology that has been developed can, in conventional systems, offer the potential of much greater circuit complexity, it offers no equivalent increase in circuit performance (*figure 1*).

This development dead-end is a direct result of the traditional approach to computer system design, which holds the view that processing is expensive compared with memory. What is known as the 'von Neuman bottleneck' has been a consequence: a single processor is connected to a vast amount of memory, and more time is spent fetching and storing data than actually processing it!

The development of VLSI technology has, however, altered the economics of system design. Nowadays, one wafer of silicon can hold 2 Mbytes of memory or 256 conventional microprocessors – we can see, therefore, that one microprocessor now costs about as much to produce as 8 kbytes of memory.

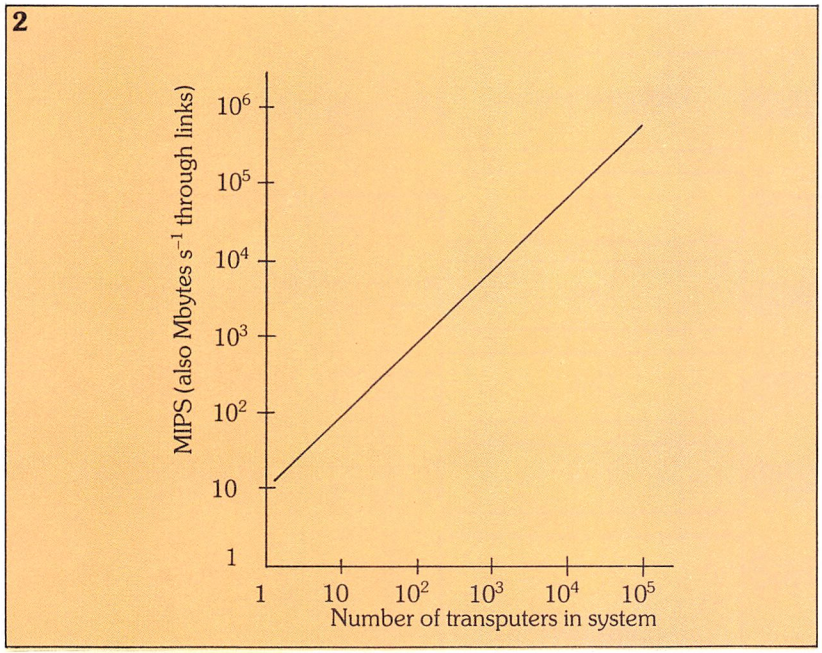
The logical next step is to design computer systems that use more than one processor, working in parallel. Once the program tasks or processes have been divided up, they can be processed in parallel (at the same time) with a subsequent improvement in processing time, and a more efficient use of memory.

**1. Conventional system performance** (measured in millions of instructions per second, or MIPS) has increased by a factor of ten each decade.




**Concurrent** programming, like this, is possible with the transputer which is designed to implement such systems. The transputer's concurrency, is derived from its specially designed language **OCCAM** which we shall consider later in the chapter.

The Japanese have predicted that computers will need one thousand times their present performance if the intelligent interaction between people and computers is to be achieved. *Figure 2* illustrates the

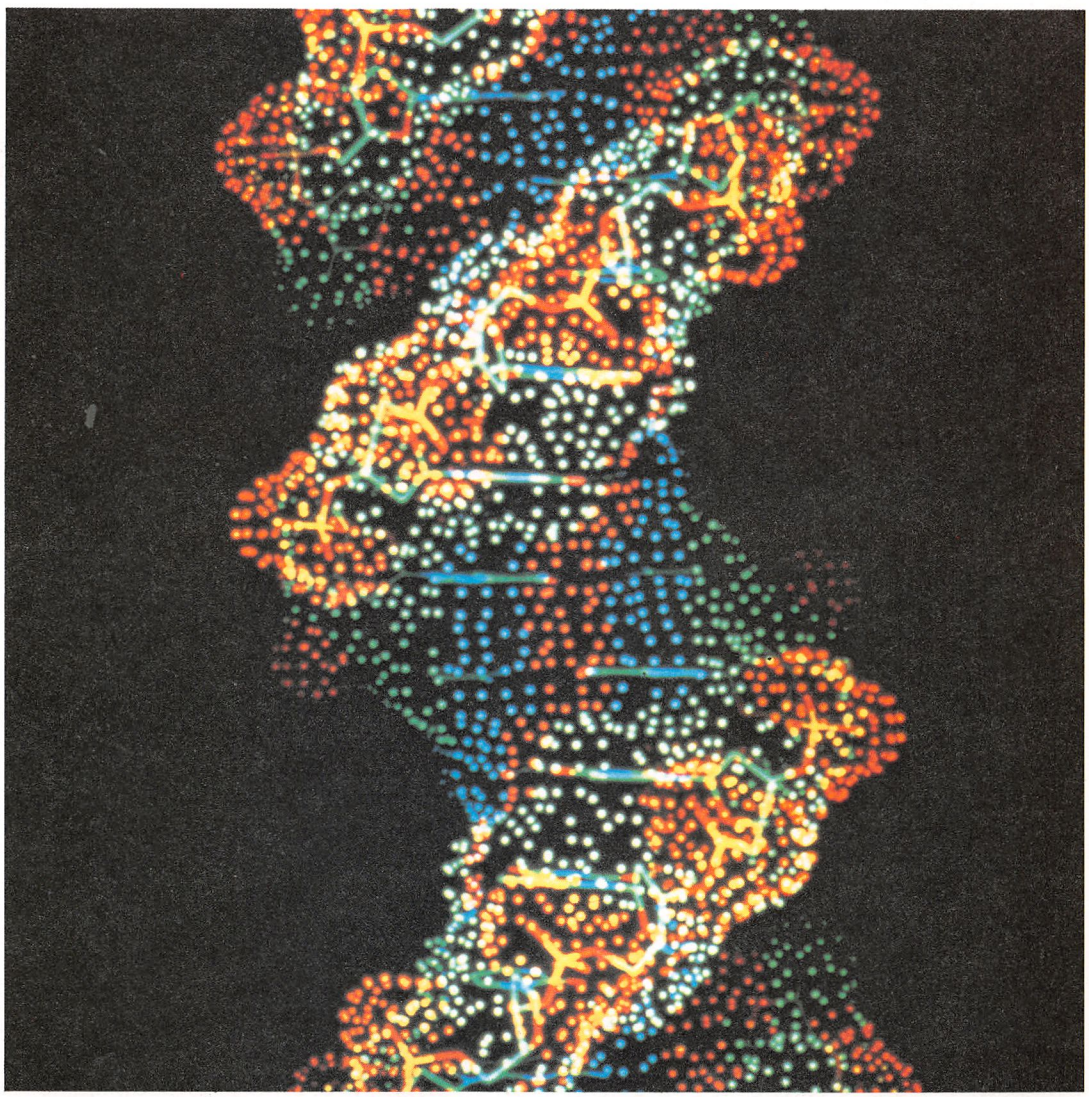



**2. Transputer system throughput.** As the number of transputers acting concurrently in a system increases, system performance improves dramatically.

way in which the number of concurrently working transputers affects the operational throughput of a computer system. As you can see, concurrency significantly improves system performance to such an extent that it probably holds the key to the development of fifth generation systems.

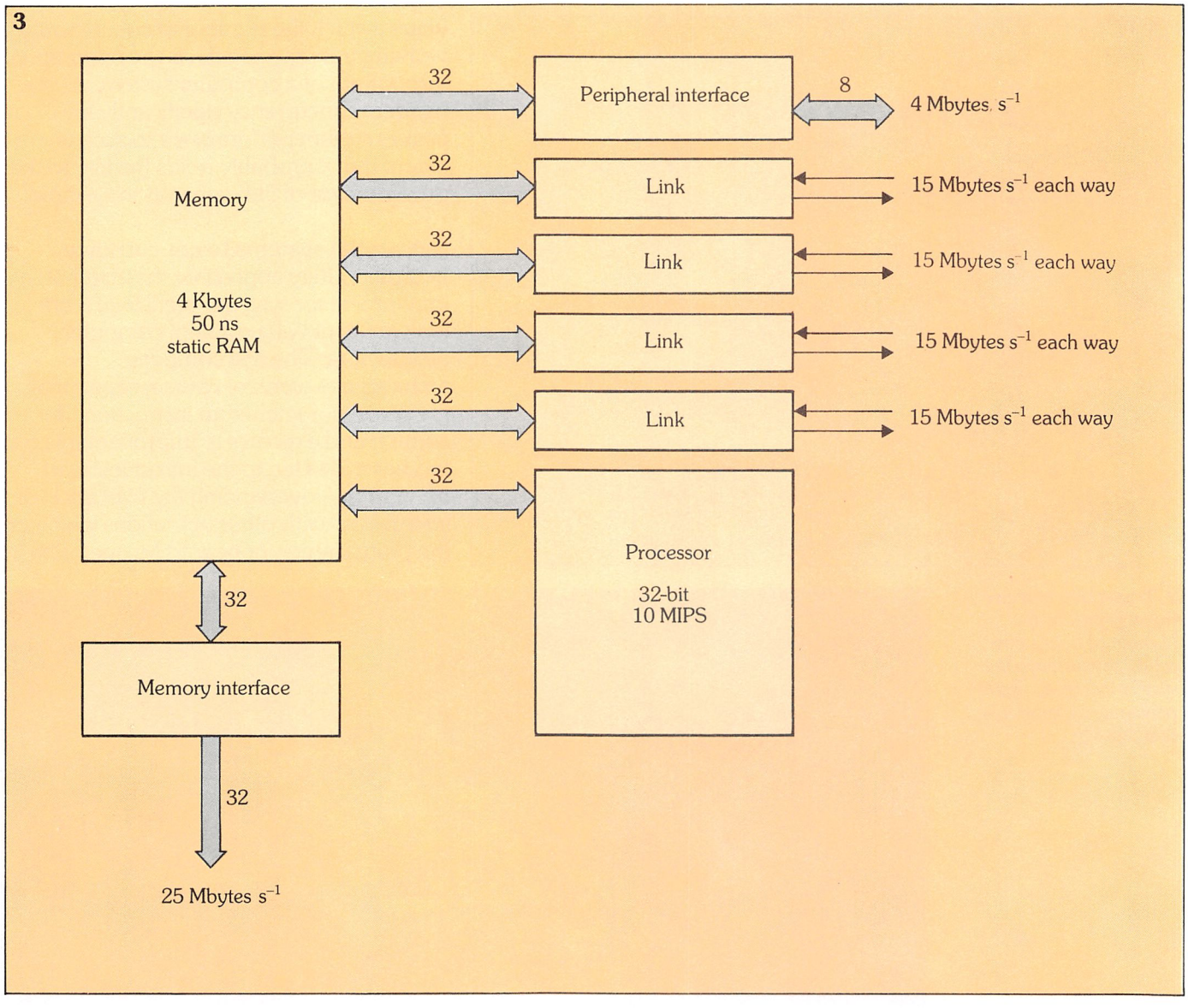
### Sequential systems to concurrency

In order to fully appreciate the effect of multiprocessor systems, we'll take a brief look at the development of computer architecture. The first computer architectures were, of course, sequential: the processor fetches an instruction, decodes it and executes it. Improvements on this basic idea led to the introduction of **pipelining** – overlapping the execution of instructions with other operations which, to the programmer, *appear* to be operating in



Science Photo Library/National Institute of Health

**Right:** computer generated image of a DNA molecule.




**3. Block diagram** of the IMS T424 transputer.

a sequential manner.

Languages and programs have, of course, exhibited certain characteristics necessary for running on sequential systems. Interrupts replaced polling which meant that a program would be stopped when a service routine had to be entered, and then resumed upon completion of the interrupt task. (Polling involved periodically operating a routine to check each peripheral, to see if servicing was required.) The resultant increase in system efficiency meant that one processor could now be shared amongst several users, probably all at remote terminals.

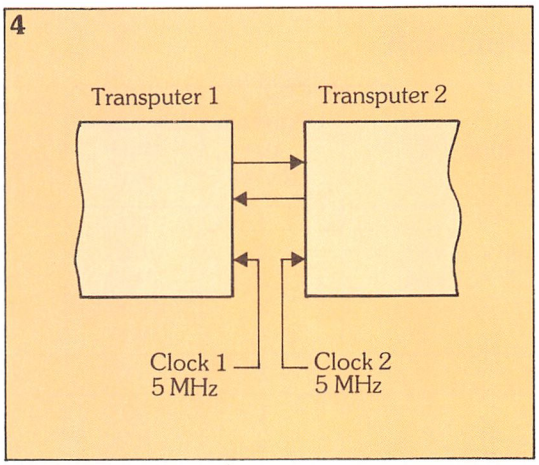
Operating systems were designed to carry out the job sharing and scheduling role and most of the early computing science effort went into designing algorithms that handled this function efficiently.

Now, as we have seen, it is economically possible and practically desirable for computer systems to have more then one processor that operates concurrently, running programs in parallel. However, the theoretical advantages of this approach also bring practical drawbacks:

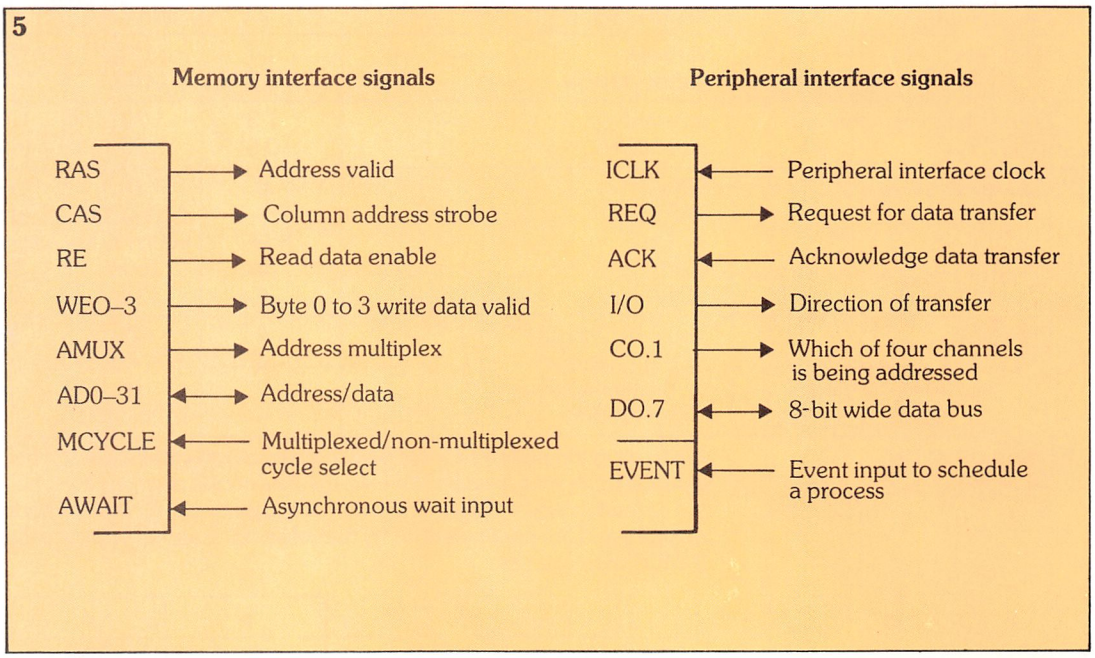
**1)** Most multiprocessor systems are designed to use a common bus, and this brings a limit to the total 'bandwidth' available between machines. Of course, the performance increase is then limited. Other constraints arise from the communications and the interrupt systems that exist between the processors.

**2)** Software, too, brings about difficulties as

**4. Clocking linked transputers.**




**5. Memory and peripheral** interface signals of the IMS T424 transputer.




there has to be some way of allowing the various parallel parts of the program to communicate with each other. As you'll appreciate, results from one part of a program can form the vital input data needed by another part. Mail-box and semaphore techniques have often been used, bringing with them attendant complexities of synchronisation which easily result in system design errors.

The transputer, though, can provide solutions to both hardware and software design problems in concurrent systems.

## Practical parallelism

The transputer overcomes the usual problems of concurrent operation in two ways: a component that can be easily added to other transputers; and it has a programming language designed especially for it.

The transputer gets around the problems of synchronism by communicating over high speed serial links. A block diagram of the IMS T424 transputer is shown in *figure 3*. As you can see, it has four communications links, so a wide variety of network types can be implemented. These links operate independently, in parallel with program execution, and provide memory-to-memory (block) transfer capability between transputers.

Each link uses two channels, which comprise an input and an output; both are used to carry data and link control information. Messages are transmitted as a sequence of bytes. The transputer's protocol arrangement that safeguards data transmission works like this: after a data byte is transmitted, the sending transputer waits to receive an acknowledgement which tells it that the receiving transputer is ready for the next byte; because transmission is so quick, the receiving transputer can transmit its acknowledgement as soon as it starts to receive a data byte, so the process is normally continuous. As you can see, this protocol ensures reliable transmission in spite of any possible delays in either transputer involved.

The speed of data transmission over each link can be set in the system's programming. The highest speed is 20

Table 1
**IMS T424 transputer performance estimation**

| Type of instruction | Operation | Program size (bytes) | Execution time (ns) | External program access† (ns) | External data access† (ns) |
|---|---|---|---|---|---|
| Arithmetic operators | +, – | 1 | 50 | p | 0 |
| | * (multiplication – 32×32=32) | 1 | 950 | 0 | 0 |
| | * (multiplication – 32×32=64) | 2 | 1950 | 0 | 0 |
| | / (division – 64/32=32) | 2 | 1950 | 0 | 0 |
| | / (remainder) | 2 | 1950 | 0 | 0 |
| Comparison operators | >, =, <>, <, <=, >= | 2 | 100 | 2p | 0 |
| Logical operators | AND, OR | 1 | 50 | p | 0 |
| | /\ , \/, ><(xor) | 2 | 100 | 2p | 0 |
| Shift operators | <<[n],>>[n] | 2 | 50n+50 | 0 | 0 |
| Identifiers | variable | 1.7 | 120 | 1.7p | d |
| | vector variable | 2.7 | 160 | 2.7p | d |
| Expression evaluation | constant | 1.3 | 70 | 1.3p | 0 |
| | parenthesis | 0.5 | 50 | 0.5p | 0.5d |
| Constructors | SEQ[n] | 0 | 0 | 0 | 0 |
| | PAR[n] | 9n–7 | 450n–200 | 7np+nd | d+4nd |
| | ALT[n] | 8n+7 | 600n+600 | 8p+2d | 5d+4nd |
| | IF[n] | 3n | 150n | 2nd | 0 |
| | WHILE | 4 | 200 | 2d | 0 |
| Primitives | !(output), ?(input) | 4 | 625 | 3p | 5d |
| | ![n], ?[n] (vector) | 4 | 50n+625 | 3p | 5d+nd |
| | := (assignment) | 0 | 0 | 0 | 0 |
| | := [n] (vector) | 4 | 100n+300 | 3p | 2nd |

† *External memory*

When program and/or data are held in external (off-chip) memory, the relevant times from the external program access and/or external data access columns must be added to the time given by the execution time column. The external access times will depend on the speed of external memory and are parameterized by the values for p and d. Typical values for p and d for some INMOS memories are:

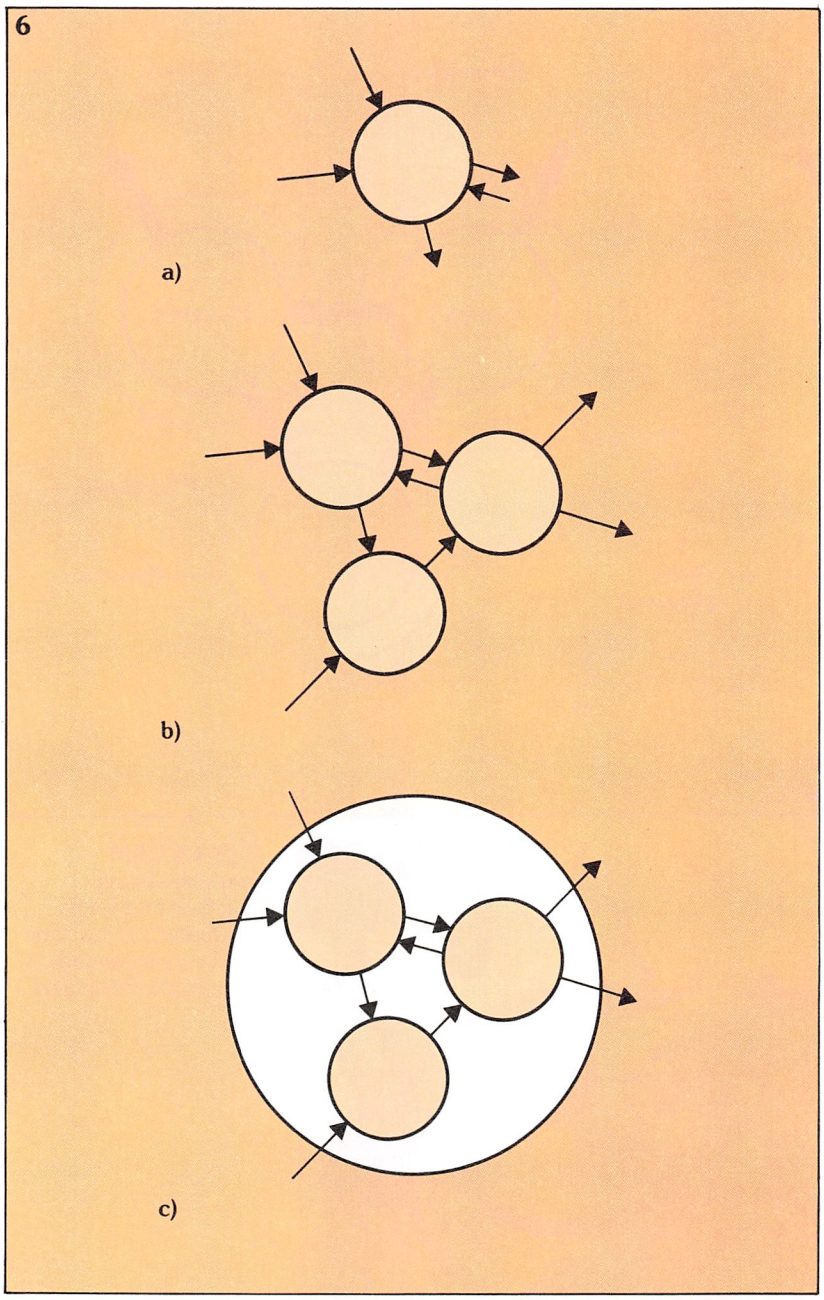
| | p | d |
|---|---|---|
| IMS 1400-70 16Kx1 SRAM | 20 | 100 |
| IMS 2600-12 64Kx1 DRAM | 40 | 150 |
| IMS 2600-15 64Kx1 DRAM | 55 | 200 |
| IMS 3630-20 8Kx8 EEPROM | 55 | 200 |

Mbits $s^{-1}$, which provides a maximum data rate of 1.8 Mbytes $s^{-1}$ on a channel. This high speed data transfer is made possible by the transputer's timing system. A low frequency clock is used to simplify the distribution of the timing signals. All transputers are designed to work from a 5 MHz signal, no matter what speed their integral performance is geared to. Once this signal is received by a transputer it is processed by clock multiplication circuitry to provide the higher frequencies needed for use on the chip.

*Figure 4* illustrates the way in which linked transputers are clocked. It is interesting to note that communication between transputers is not critically dependent on frequency. This means that transputers with independent clocks can reliably communicate with each other.

The transputer's memory interface can handle mixed memory systems. That is to say, it generates signals for both multiplexed and non-multiplexed memory, the timing of which is controllable by external signals.

The T424's memory interface is a 32-bit multiplexed address and data bus,




**6. (a) A single process; (b)** the concurrent system of processes; **(c)** a collection of processes is itself a process. In this way, processes may have internal concurrency.

which allows up to 4 Gbytes of memory to be connected – including its 4 kbytes of on-chip memory. The use of on-chip memory gives a significant advantage in speed and power consumption. On-chip memory gives a cycle time of 50 ns, but the program cannot tell any difference between on and off chip memory, apart from performance. Static RAMs and ROMs, and dynamic RAM can be supported by the transputer. The transputer's memory and peripheral interface signals are shown in *figure 5*.

## The transputer range

The 32-bit IMS T424 is the first in a proposed family of INMOS transputers; a 16-bit device, the IMS T222, is planned to be the second. The instruction sets for these two transputers will be identical, and programs written for one will be able to run on the other, providing the 16-bit arithmetic and address space range of the IMS T222 is not exceeded. Both transputers run at an average rate of 10 MIPS (millions of instructions per second). The IMS T424's estimated performance over its range of functions is shown in *table 1*.

## The processor

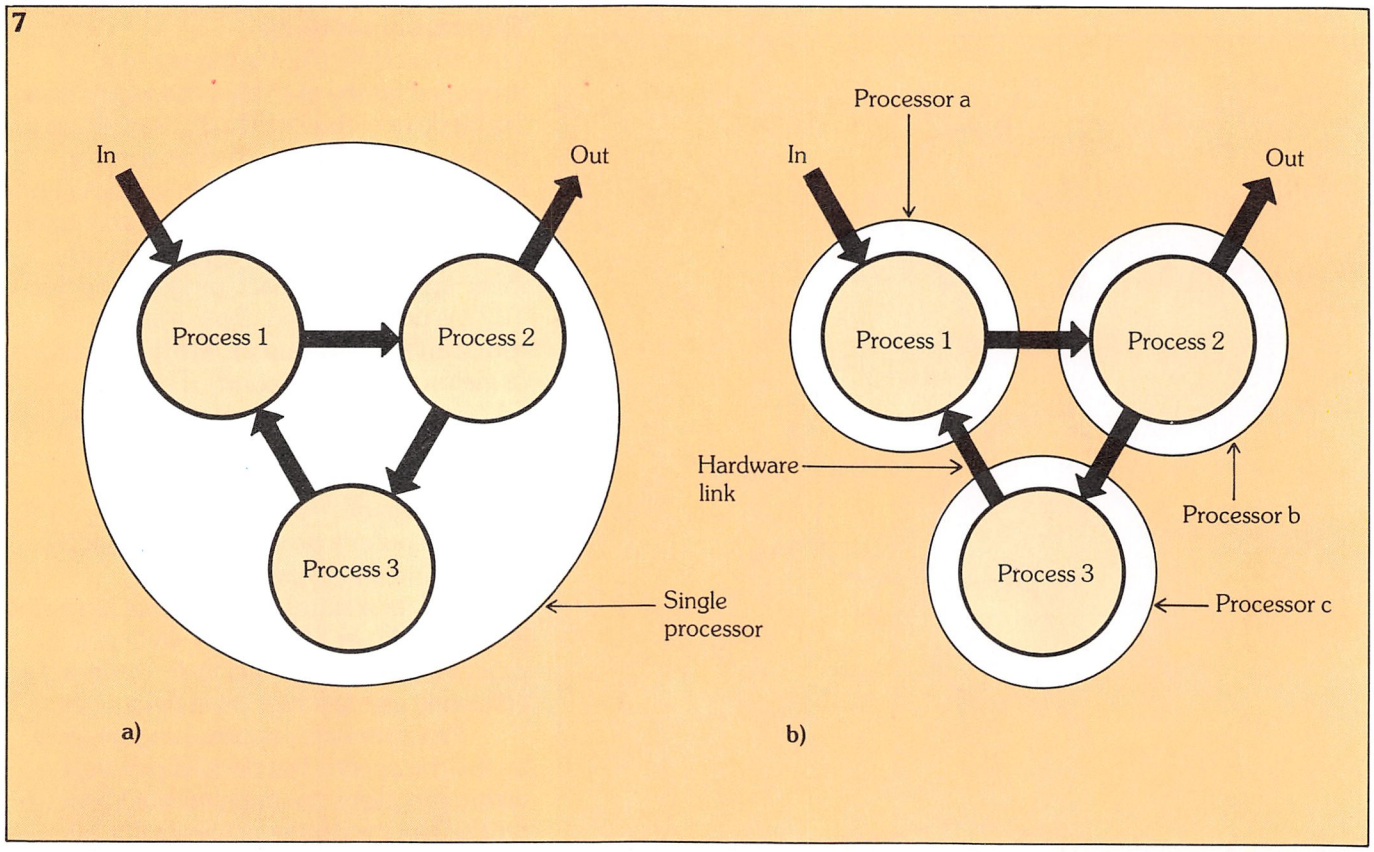
The transputer's processor will either be programmed in its own language, OCCAM, or a standard high level language like PASCAL or FORTRAN. In fact, the processor has been specifically designed to efficiently execute high level languages.

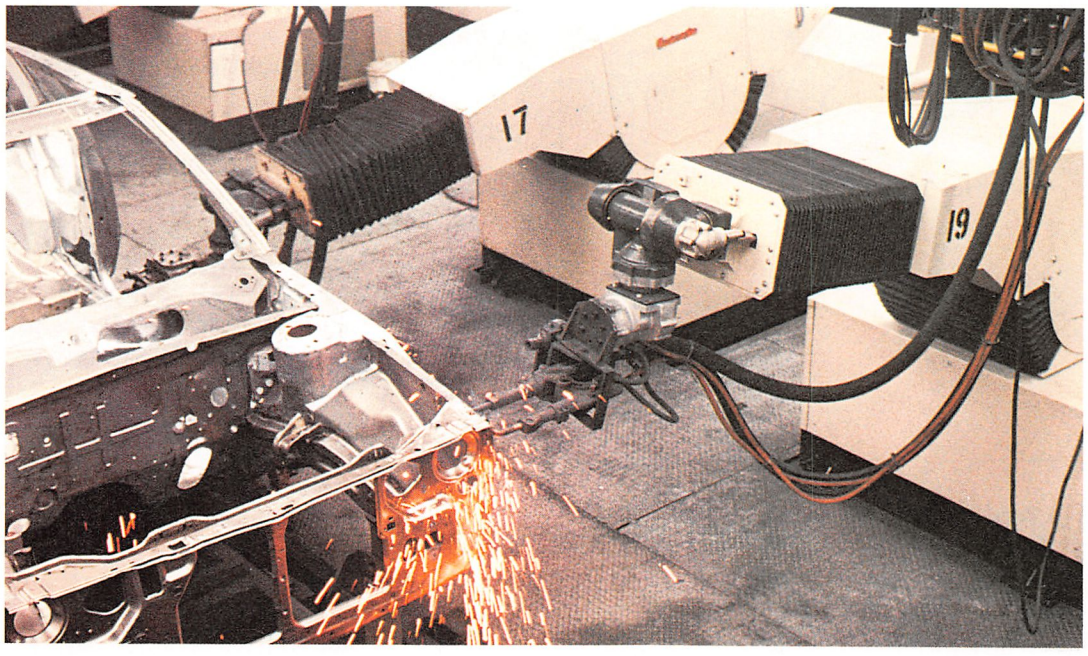
Programs are executed *sequentially* by the transputer, but parallel processes within programs are implemented *concurrently*. The transputer achieves this by *sharing* its time between the sets of processes that are **active** at any one moment. An active process is one that is to be run, and is not waiting for any input or output to be carried out.

The process under execution runs until it has to wait for further input – it is then set as **inactive** and the next process in the active queue is executed. When a communication is made to a channel which is ready, the message is passed, and the waiting process is then linked to the end of the active process queue while the current process continues execution.

The transputer directly implements the process model of computing. A process is defined as an independent computation, with its own program and data, which can communicate with other processes that are under execution at the same time (*figure 6a*). Communication between processes is accomplished by passing messages along explicitly defined channels (*figure 6b*).

As well as being designed to implement a set of concurrent processes, the transputer's external behaviour also corresponds to the formal model of a process. This means that systems with many interconnected transputers can be created –




Science Photo Library/Tom McHugh

**7. (a) OCCAM processes** being executed on a single transputer; **(b)** the same processes implemented on multiple transputers.

**Left:** microprocessor controlleed robots automatically welding cars on the production line at Chrysler Motors, Missouri, U.S.

each transputer executing a set of processes.

As a program is defined as a set of processes, it can be mapped onto such a system in many ways, to minimise cost, optimise throughput, or to maximise the responsiveness to specific events. *Figure 7a* shows a program's processes being implemented on a single transputer, while in *figure 7b*, the same processes are implemented by multiple transputers. This change in the nature of the processing can be directly implemented, without changing the software at all.